# DATA-DRIVEN ADAPTIVE LEARNING SYSTEMS

BY

XIAO LI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Educational Psychology
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Jinming Zhang, Chair
Professor Hua-hua Chang
Professor Carolyn J. Anderson
Assistant Professor Justin Kern

## ABSTRACT

Adaptive learning systems are capable of providing more adaptive and efficient assessment and learning experiences for learners than traditional classroom settings. A conventional adaptive learning system involves a learner, a latent trait estimator, and a learning strategy/plan. The latent trait estimator measures the learner's latent traits from his/her responses to the test, where computerized adaptive testing (CAT) or computerized classification testing (CCT) tailors test items to learners' abilities so as to give a more efficient latent trait estimation. On the other hand, the learning plan (called policy) is another key component of such systems. It is the algorithm that designs the learning paths, or in other words, selects learning materials for learners based on the information such as the learners' current progresses and skills, learning material contents. In this thesis, we discuss and address issues related to the adaptive test and learning problems using data-driven methods.

In the first chapter, we discuss the challenge of content balancing in variable-length adaptive tests and propose feasible data-driven methods. Content balancing is one of the most important issues in CCT. To adapt to variable-length forms, special treatments are needed to successfully control content constraints without knowledge of the test length during the test. To this end, we propose the concept of "look ahead" and "step size" to adaptively control content constraints in each item selection step. The step size gives a prediction of the number of items to be selected at the current stage, that is, how far we will look ahead. Two look-ahead content balancing (LA-CB) methods, one with a constant step size and another with an adaptive step size, are proposed as feasible solutions to balancing content areas in variable-length computerized classification testing (VL-CCT). The proposed LA-CB methods are compared with conventional item selection methods in variable-length tests under different classification methods' settings. Simulation results show that integrated with heuristic item selection methods, the proposed LA-CB methods outperform the conventional item selection methods with fewer constraint violations and higher classification accuracy.

The second issue we address is to find the learning policy that designs the optimal learning path in an adaptive learning system under hierarchical skill

structures. To this end, we first develop a model for learners' hierarchical skills in the adaptive learning system. Based on the hierarchical skill model and the classical cognitive diagnosis model, we further develop a framework to model various levels of proficiency related to hierarchical skills. The optimal learning policy in consideration of the hierarchical structure of skills is found by applying a data-driven algorithm-reinforcement learning method, which does not require information about learners' learning transition processes. The effectiveness of the proposed framework is demonstrated via simulation studies.

Lastly, we solve the problem of finding a learning policy assuming latent traits to be continuous with an unknown transition model. We formulate the adaptive learning problem as a Markov decision process (MDP). We apply a model-free deep reinforcement learning algorithm—the deep Q-learning algorithm—that is data-driven and can effectively find the optimal learning policy from data on learners' learning process without knowing the actual transition model of the learner's continuous latent traits. To efficiently utilize available data, we further develop a transition model estimator that emulates the learner's learning process using neural networks. The transition model estimator can be used in the deep Q-learning algorithm so that it can more efficiently discover the optimal learning policy for a learner. Numerical simulation studies verify that the proposed algorithm is very efficient in finding a good learning policy, especially with the aid of a transition model estimator, it can find the optimal learning policy after training using a small number of learners.

*To My Family.*

## ACKNOWLEDGMENTS

I would first like to extend my most sincere gratitude to my advisors Dr. Jinming Zhang and Dr. Hua-Hua Chang for their generous guidance and endless support throughout my graduate study at University of Illinois at Urbana-Champaign. I am grateful to Dr. Jinming Zhang for supporting me in every aspect through the most difficult times in my Ph.D. studies. I have benefited a lot by learning from his creative thinking, scientific and academic rigour, personality of integrity, and caring for others. I also want to thank Dr. Hua-Hua Chang for his tremendous guidance on my research and introducing me to various opportunities. Without his kind consideration and patient help in both my work and my life, and learning from his philosophy of taking challenges optimistically, I cannot grow so much professionally and personally. My advisors were and remain my role models for a scientist and mentor.

My sincere thanks also go to Dr. Carolyn Anderson and Dr. Justin Kern for their precious time to serve on my Doctoral committee, for teaching me and sharing their expertise knowledge and experiences as a scientist, for their patience in answering my questions and discussing on my research, for their inspiring encouragements. It's my great pleasure to work with them, present my work to them and listen to their valuable feedbacks.

This work wouldn't be possible without many others' endless supports. I would express many thanks to my family and all my friends for the encouragements and joy they have brought to me.

Lastly, my deepest gratitude and love goes to my grandparents, my parents Qun Li and Wenping Fan, and my boyfriend Hanchen Xu. Your unconditional supports, comforts, encouragements and love give me the strength to walk through all the difficult moments and accompany me to enjoy all the joyful moments in my life.

## TABLE OF CONTENTS

# CHAPTER 1

# LOOK-AHEAD CONTENT BALANCING METHODS IN VARIABLE LENGTH COMPUTERIZED CLASSIFICATION TESTING

## 1.1 INTRODUCTION

The computerized classification testing (CCT) method has been applied in a variety of proficiency tests, to classify examinees into two or more mutually exclusive groups (Parshall, 2002). Different from the computerized adaptive testing (CAT) method with respect to point estimation of ability, the CCT method does not necessarily acquire an accurate estimation of ability values (Weiss and Kingsbury, 1984; Thompson and Prometric, 2007).

For the purpose of further improving test efficiency, variable-length computerized classification testing (VL-CCT) is adopted (Swygert, 2002; Thompson and Prometric, 2007). Variable-length testing refers to tests in which not all examinees receive the same number of items. Before a pass/fail decision is made, an examinee with high or low ability that is far from the cutoff score will receive a relatively small number of items compared to an examinee with ability closer to the cutoff score.

The purpose of the VL-CCT method is to provide the decision with as few items as possible, while maintaining decision accuracy at a certain level. The VL-CCT method is a powerful and efficient approach to classify examinees into groups using variable test lengths adapted to abilities. It outperforms fixed-length tests in at least three aspects: 1) offering substantially shorter tests than a conventional fixed-length test while maintaining a similar level of classification accuracy (Kingsbury and Weiss, 1983); 2) conforming to the "equal measurement error variance" with fixed standard error of measurement (SEm) as a stopping rule (Huo, 2009); and 3) allowing subsequent statistical analyses involving measurement errors easily handled (Thissen and Mislevy, 2000; Wainer et al., 2000).

Currently, the VL-CCT method is not as widely adopted as the fixed-length method in educational and psychological assessments for several reasons. First, it is reported that extremely short tests can affect examinees' fairness perceptions (Tonidandel et al., 2002; Huo, 2009). Second, it is dif-

ficult to incorporate all statistical and non-statistical constraints into a VL-CCT design. In the VL-CCT implementation, constraints include content balancing, exposure control, answer key balancing, etc. Content balancing refers to the case when a certain proportion of items needs to be selected from each content area. Exposure control means item exposure rate should be retained under a specific threshold. Ideally, items should not be over-exposed or under-exposed, in order to protect test security and maximize item pool usage. Answer key balancing stands for correct answers should be uniformly distributed among options (Sympson and Hetter, 1985; Chang and Ying, 1999; Cheng and Chang, 2009). However, as the total number of administered items is unknown before a VL-CCT test is terminated, traditional item selection methods cannot accommodate non-statistical constraints properly without pre-specifying a content area range.

The importance of content balancing has been demonstrated by many researchers (Green et al., 1984; Thissen and Mislevy, 2000; Wainer et al., 2000). A number of methods have been proposed to manage non-statistical constraints including the constraint CAT method (CCAT; Kingsbury and Weiss, 1983), the modified multinomial model method (MMM; Chen and Ankenman, 2004), the modified CCAT method (MCCAT; Leung et al., 2000), the maximum priority index method (MPI; Cheng and Chang, 2009), and the content weighted item selection index method (CWI; Huo, 2009). Among them, the CWI method can be adapted to accommodate to constraint management in variable-length tests. Furthermore, the MPI method was adjusted and introduced in variable-length multidimensional CAT (Yao, 2013; Su, 2015, 2016). However, it is still a challenging task to control all constraints simultaneously in a variable length test setting. Thus, demand exists to develop new content balancing methods that are specifically designed for variable-length tests.

The first chapter addresses these challenges by proposing two feasible methods based on a new design, named look-ahead content balancing (LA-CB), that gains control over content coverage in severely constrained VL-CCT programs. The concepts of "look ahead" and "step size" are proposed here which aim at controlling content constraints in each item selection step, while the step size, indicating how far to look ahead, is adopted to predict the number of items to be selected at the current stage. Integrated with the MPI item selection method, the two LA-CB based methods simultaneously

accommodate non-statistical constraints in VL-CCT. Beyond that, these LA-CB methods are easy to implement in VL-CCT tests. The LA-CB methods are then compared with the MPI and CWI methods with respect to their performance in constraint management and classification accuracy.

## 1.2 METHODS

The three-parameter logistic model (3PLM) (Hambleton and Swaminathan, 2013) is mostly frequently used in CCT programs. The 3PLM defines the probability that an examinee with ability $\theta$ answering item $j$ correctly as

$$P_j(X = 1|\theta) = c_j + (1 - c_j)\frac{e^{a_j(\theta - b_j)}}{1 + e^{a_j(\theta - b_j)}}, \tag{1.1}$$

where $a_j$ is the item discrimination parameter, $b_j$ is the item difficulty parameter and $c_j$ is the guessing parameter or a lower asymptote.

One of the most widely used item selection methods in CCT programs is the maximum Fisher information method (Lord, 1980; Wainer et al., 2000). It selects the next item with the maximum value of Fisher information evaluated at current ability estimate point $\hat{\theta}$. The Fisher information in the 3PLM is expressed as

$$I_j(\hat{\theta}) = \frac{(1 - c_j)a_j^2 e^{-a_j(\hat{\theta} - b_j)}}{[1 + e^{-a_j(\hat{\theta} - b_j)}]^2\{1 - c_j + c_j[1 + e^{-a_j(\hat{\theta} - b_j)}]\}}. \tag{1.2}$$

Other than the maximum Fisher information method, the MPI method measures both the information each item carries, and each item's contribution toward meeting constraints.

### 1.2.1 Content Balancing Item Selection Methods

#### Maximum Priority Index

The MPI method is a flexible item selection algorithm that incorporates content balancing constraints in fixed-length CAT. The MPI method heuristically balances constraints in an item selection procedure by including a multiplier in front of an item's Fisher information that quantifies the contribution to $\theta$ estimation. Clearly, the larger the MPI, the more attractive the

item is to be administered. The priority index of item j is defined as:

$$PI_j = I_j(\hat{\theta}) \prod_{k=1}^{K} (\omega_k f_k)^{c_{jk}}, \tag{1.3}$$

where $I_j$ represents the Fisher information of item j with regards to $\hat{\theta}$ and $f_k$ is the scaled remaining quota of constraint $k$ which is defined later. The $c_{jk}$ is the indicator of whether constraint $k$ is relevant to item j, where $c_{jk} = 1$ if $k$ is relevant, and 0 otherwise. The $\omega_k$ is a predefined weight regarding constraint $k$, which is used to quantify the importance of content constraints; that is, major content constraints will receive large weights.

Suppose that the target number of items dealing with specific content constraint $k$ is $X_k$ and that the number of such selected items is $x_k$. The resulting scaled quota $f_k$ is

$$f_k = \frac{X_k - x_k}{X_k}. \tag{1.4}$$

The $X_k$ can vary over different item selection phases as well as constraints. For example, if content area $k$ involves a lower bound $l_k$ and an upper bound $u_k$, $X_k$ is equal to $l_k$ in the first phase and is $u_k$ in the second. In variable-length tests, the upper limit is bounded by a ratio $\tilde{u}_k\%$ and the maximum test length $U$, which gives $u_k = U \times \tilde{u}_k\%$.

To ensure adequate items administered to examinees and a resulting reliable test, most VL-CCT programs set both lower and upper bounds on total test length, as well as content area constraints. Therefore, the lower bound of each content area is handled in the first phase and the upper bound in the second phase.

In particular, the desired exposure rate $r$ can be treated as an upper limit in $f_k$ for exposure control purposes, which is expressed as

$$f_{kr} = \frac{r - n/N}{r}, \tag{1.5}$$

where $r$ is the exposure rate upper limit, $n$ is the frequency that item $j$ has been administered, and $N$ is the total number of examinees.

Content Weighted Item Selection Index

One content balancing control method proposed for variable-length tests is content weighted item selection index method (CWI) (Huo, 2009). The method incorporates adapted a-stratified methods to control content constraints in variable-length CAT.

Let $l_k$ and $u_k$ denote the lower and upper bounds of the constraint $k$ respectively, and let $x_k$ denotes the number of selected items from the constraint $k$. The CWI method is calculated in two phases. In the first phase, the index is expressed as

$$CWI = \frac{l_k}{l_k - x_k + 1} * |\hat{\theta} - b|. \tag{1.6}$$

In the second phase, the index is

$$CWI = \frac{u_k}{u_k - x_k + 1} * |\hat{\theta} - b|. \tag{1.7}$$

To adjust the CWI method in the variable-length setting with exposure control, the author proposes an adapted a-stratified method. The method selects items from strata in a circularly increasing or decreasing order in the second phase, instead of in a strictly ascending or descending order from the original a-stratified item selection method (Chang and Ying, 1999). This adapted method achieves the best result among other adaptations in the original paper. Therefore, we will continue to use the CWI with the adapted a-stratified method as one of reference methods to compare with the new methods presented below in our simulation studies.

Look-ahead Content Balancing

The problem of using existing content balancing methods is that the upper bound of each content area is unknown before the test is terminated. In VL-CCT programs, each content balancing constraint usually includes both a lower bound and an upper bound. The lower bounds are fixed values to ensure adequate items administered to examinees and the reliability of the test. The upper bounds are usually controlled by a target percentage. As a result, when the total test length is changing, the program cannot determine the exact upper bounds. The existing MPI method and the CWI

method are both using maximum upper bounds, which are the total test length multiplied by the target percentages, as the target upper bounds in the second phase. However, the maximum upper bounds can be much larger than the actual ones since some tests may terminate early. As a result, the content constraints cannot be controlled properly.

To solve this problem, we first proposed a straight forward solution. The upper bounds are decided by a fixed value named "step size". By looking one step ahead, the upper bounds keep determined by the existing number of selected items plus the step size in each item selection procedure. An alternative method is to determine the step size by a confidence interval derived from the Fisher information. The upper bounds are then decided in the same way.

We introduced the idea of looking ahead by taking one step forward in both methods. Both of them prove to be reliable in maintaining high test accuracy and content management. In addition, we can use the flexible values of step size to decide the priority of achieving higher classification accuracy or fewer constraint violations. Besides, the Fisher information contributes the measure which further refines the step size's precision in determining upper bounds. The resulting VL-CCT program shows its high test efficiency over fixed-length tests without compromising constraint management.

Specifically, the LA-CB design adopts the idea of two-phase item selection strategy (Cheng et al., 2007; Cheng and Chang, 2009). It handles lower bounds in the first phase and upper bounds in the second. Same as notations above, let $x_k$ denotes number of selected items from content area $k$. Equations below must be satisfied:

$$x_k \geq l_k, \tag{1.8}$$

and

$$x_k \leq TL * \tilde{u}_k\%, \tag{1.9}$$

where $l_k$ is the lower bound for content area $k$, $\tilde{u}_k\%$ is the target percentage of content area $k$ in the second phase and $TL$ is total test length. The priority index $PI_j$ is then computed by (1.3).

In the first phase, we have:

$$f_k = \frac{l_k - x_k}{l_k}. \tag{1.10}$$

So the $f_k$ gives the quota of the distance between the lower bound and the current selection length.

In the second phase, because both the total test length and the total number of items received by examinees are changing, we should have a solution to determine what would be the remaining length. The way to go about it is to take one step ahead, by introducing either a constant value or an adaptive value determined by the confidence interval. We call the value step size $S$. Suppose the maximum test length is $U$ which is larger than or equal to the actual test length $TL$. Then the target percentage $\tilde{u}_k\%$ must satisfy

$$x_k + S * \tilde{u}_k\% \leq U * \tilde{u}_k\%, \tag{1.11}$$

which gives:

$$1 \leq S \leq U - \sum_{k=1}^{K} x_k. \tag{1.12}$$

Inequalities (1.11) and (1.12) indicate that the number of selected items plus $S$ cannot exceed maximum test length in VL-CCT programs. Besides, if the test is still in progress (i.e., maximum test length has not been reached and the termination criteria has not been satisfied), at least one item should be selected, in which case the value of $S$ is at least 1. Therefore, the step size $S$ can be a constant integer number within the range given by (1.12). We name the LA-CB method with constant step size $S^{\text{constant}}$ as LA-CB-C.

To further improve the precision in determining the upper bound, we used the ability confidence interval method (ACI) to predict the step size $S$. By evaluating the distance of current Fisher information and the desired Fisher information, the value of $S$ is calculated. As a result, constraints under each content area can better controlled. The method is denoted as LA-CB-A.

By the ACI method, a confidence interval (CI), based on $\hat{\theta}$ and the conditional standard error of measurement $SEm$, will be constructed and compared to the cut score. The $(1-\alpha)\%$ CI is expressed as:

$$\hat{\theta} - Z_\alpha * \text{SEm} < \theta < \hat{\theta} + Z_\alpha * \text{SEm}, \tag{1.13}$$

where $Z_\alpha$ is the cutoff point of the standard normal distribution.

To estimate $SEm$, by central limit theorem, under local independence and large $n$ assumptions, we have:

$$\text{SEm}(\hat{\theta}) \to \frac{1}{\sqrt{\Sigma_{j=1}^{n} I_j(\theta)}},\ as\ n \to \infty, \tag{1.14}$$

After the first $J$ items have been administered, CI is approximated by:

$$\hat{\theta} - Z_\alpha * \frac{1}{\sqrt{\Sigma_{j=1}^{J} I_j(\hat{\theta})}} < \theta < \hat{\theta} + Z_\alpha * \frac{1}{\sqrt{\Sigma_{j=1}^{J} I_j(\hat{\theta})}}. \tag{1.15}$$

where $I_j(\hat{\theta})$ represents Fisher information of item j evaluated at the ability estimate point $\hat{\theta}$. Since the LA-CB-A method is applied in the second phase, at least $l_k$ items have already been administered. The number of items is large enough so the accumulated Fisher information can be used to approximate $SEm$.

Denote the cutoff score as $\theta_0$. If the lower bound of the CI is equal to $\theta_0$, the entire CI will lie on the right side of $\theta_0$, leading to the classification of passing the test under ACI method, where

$$\theta_0 = \hat{\theta} - Z_\alpha * \frac{1}{\sqrt{FI_0}}. \tag{1.16}$$

On the contrary, if the upper bound of the CI is equal to $\theta_0$, the entire CI will lie on the left side of $\theta_0$, which gives

$$\theta_0 = \hat{\theta} + Z_\alpha * \frac{1}{\sqrt{FI_0}}. \tag{1.17}$$

The test will terminate and the examinee will be classified as failing the test.

Both equations (1.16) and (1.17) give the same total desired Fisher information $FI_0$ to terminated test which is calculated by:

$$FI_0 = [\frac{Z_\alpha}{\theta_0 - \hat{\theta}}]^2. \tag{1.18}$$

Therefore, the number of items to be selected in next steps has the range

$$\frac{FI_0 - \sum_{j=1}^{J} I_j(\hat{\theta})}{\max(I_{\text{unselected}})} < \#\text{ items} < \frac{FI_0 - \sum_{j=1}^{J} I_j(\hat{\theta})}{\min(I_{\text{unselected}})}, \tag{1.19}$$

where $\max(I_{\text{unselected}})$ and $\min(I_{\text{unselected}})$ represent the maximum and minimum Fisher information based on current $\hat{\theta}$, respectively, for an item in the remaining item pool.

To conservatively control content constraints, the predicted number of remaining items should be as small as possible. Therefore, the LA-CB-A method uses the left bound in (1.19) as the look-ahead upper bound. The adaptive step size is calculated as

$$S_0^{\text{adaptive}} = \frac{FI_0 - \sum_{j=1}^{J} I_j(\hat{\theta})}{\max(I_{\text{unselected}})}. \tag{1.20}$$

When the test is in a relatively early stage, the standard error of an estimated ability is large and the accumulated Fisher information is not close to $FI_0$ yet. As a result, the adaptive step size $S_0^{\text{adaptive}}$ can be very large. To take advantage of $S_0^{\text{adaptive}}$ while having it controlled in a reasonable range, we integrated the $S_0^{\text{adaptive}}$ with the constant step size $S^{\text{constant}}$. The resulting $S^{\text{adaptive}}$ in the LA-CB-A method is calculated by

$$S = \max\{1, \min\{S_0^{\text{adaptive}}, S^{\text{constant}}\}\}. \tag{1.21}$$

With the step size $S$ for either the LA-CB-C or LA-CB-A method, the quota $f_k$ is calculated by

$$f_k = \frac{(\sum_{k=1}^{K} x_k + S) * \tilde{u}_k\% - x_k}{(\sum_{k=1}^{K} x_k + S) * \tilde{u}_k\%}. \tag{1.22}$$

The priority index is calculated by (1.3) for each item $j$ and the item with the maximum priority index is selected and administered.

$FI_0$, $S$, and $f_k$ are predicted and updated after each item is answered and items are administered following the same procedure until termination

criterion is satisfied or the maximum test length is reached. Examinees are classified as pass/fail based on the classification criterion if the test terminates before the maximum test length is reached. Otherwise, examinees are classified based on the comparison between the estimated ability $\hat{\theta}$ and the cut score $\theta_0$.

### 1.2.2 Classification Methods

#### Sequential Probability Ratio Test

The sequential probability ratio test (SPRT) (Wald, 1973; Eggen, 1999) has proven to be a reliable method in the adaptive test for classifying examinees into categories (Spray and Reckase, 1996; Eggen and Straetmans, 2000). It compares the ratio of the likelihoods of two competing hypotheses. In CCT programs, the likelihood is calculated with the probability of an examinee's response to item $i$ given the true hypothesis. The probability is calculated with an IRT item response function.

To achieve this approach, the statistical hypotheses are formulated as

$$H_0 : \theta \leq \theta_0 - \delta = \theta_1, \tag{1.23}$$

against

$$H_1 : \theta \geq \theta_0 + \delta = \theta_2, \tag{1.24}$$

where $\delta$ is the indifference zone, accounting for the uncertainty of decisions due to measurement error. The value $\theta$ is close to the true ability measure $\theta_0$.

Acceptable decision error rates are then specified as:

$$P(\text{retain } H_0 | H_0 \text{ is true}) \geq 1 - \alpha, \tag{1.25}$$

and

$$P(\text{retain } H_0 | H_1 \text{ is true}) \leq \beta, \tag{1.26}$$

where $\alpha$ and $\beta$ are nominal Type I and Type II error rates, respectively.

Tests meeting these decision error rates are then implemented using the SPRT. The test statistic used is the ratio between the values of the likelihood

functions under the alternative hypothesis and the null hypothesis.

$$LR(\theta_2, \theta_1; \mathbf{y}) = \frac{L(\theta_2; \mathbf{y})}{L(\theta_1; \mathbf{y})} = \frac{\prod_{j=1}^{K} P_j(\theta_2)^{y_j}[1 - P_j(\theta_2)]^{1-y_j}}{\prod_{j=1}^{K} P_j(\theta_1)^{y_j}[1 - P_j(\theta_1)]^{1-y_j}}, \tag{1.27}$$

where $\mathbf{y}$ denotes responses $y_1, y_2, ..., y_K$ and $K$ denotes the total number of items. $P_j(\theta)$ is the item response function of the 3PLM from equation (1.1). Large values of this ratio indicate that the examinee's $\theta$ is above $\theta_0$, and small values indicate that $\theta$ is below $\theta_0$. That is, a statistical test satisfies acceptable decision error rates if it uses the following procedure (Eggen, 1999):
if

$$\frac{\beta}{1-\alpha} < LR(\theta_2, \theta_1; \mathbf{y}) < \frac{1-\beta}{\alpha}, \tag{1.28}$$

the sampling procedure continues; if

$$LR_k(\theta_2, \theta_1; \mathbf{y}) \leq \frac{\beta}{1-\alpha}, \tag{1.29}$$

we accept $H_0$ and classify the examinee as failing in the test; if

$$LR_k(\theta_2, \theta_1; \mathbf{y}) \geq \frac{1-\beta}{\alpha}, \tag{1.30}$$

we reject $H_0$ and decide that the examinee pass the test.

Ability Confidence Interval

The ACI method is an alternative way to make a classification decision. A 95% CI is constructed around the examinee's estimated theta after each item administered. If the examinee's 95% CI is above the cut score $\theta_0$, then the examinee passes the test. If the CI falls below $\theta_0$, then the examinee fails. If $\theta_0$ is equal to or within the examinee's CI, then the test will continue until a pass/fail decision can be made or the maximum test length is reached.

## 1.3 SIMULATION STUDIES AND RESULTS

### 1.3.1 Overview

Three simulation studies were conducted. In the first simulation study, the evaluation between the ACI and SPRT methods is based on classification accuracy and test efficiency criteria in the application of the LA-CB-C method. The main purpose of the first study is to choose a preferable classification method in the current setting so that the preferred one would be applied in the following two simulation studies. Only the results of LA-CB-C method are presented here since the LA-CB-A method produces similar results. In the second simulation study, we evaluate whether the LA-CB-C method controls content constraints better than the existing MPI method and the CWI method in VL-CCT tests, where baselines are taken to be MFI without exposure control and the randomized method. The comparisons are conducted with respect to multiple perspectives including classification accuracy, test efficiency, content balancing and exposure control. In the third simulation study, we examine whether the LA-CB-A method further improves the content balancing performance on top of the LA-CB-C method. Details of the settings and the results of the three studies are discussed in the following subsections.

### 1.3.2 Data Generation

#### Item Pool Structure

A hypothetical item bank is simulated under the 3PLM with 400 items, partitioned into 4 stages with parameter-a evenly distributed at 0.5, 1.0, 1.5, 2.0. Other item parameters are generated as $b \sim N[0, 1]$ and $c \sim U[0, 0.25]$. The item bank is evenly divided into 4 content areas, each of which contains 100 items. Each content area is assumed with a target percentage of 25%. The 4 contents are considered equally important and the weights are all set as 10. The minimum and maximum test lengths are set at 28 and 60. Therefore, for each content area, the number of selected items under each constraint $k$ ($k = 1, 2, 3, 4$) should be bounded between integers 7 and 15.

As for test security purpose, the exposure rate of all items is required to be controlled under 0.2, which means items are administered to no more than

20% examinees. The constraint is expressed in equation (1.5). Because the simulated test is considered high-stake, the weight of the exposure control constraint is set to be 100.

Examinee Generation

We drew 2,000 $\theta$s from $N[0,1]$ as our simulated examinees. To mitigate the randomness in the results, twenty replications were performed for each of the 18 step sizes of the LA-CB-C and LA-CB-A methods, and for each of the other four item selection methods, using the same item bank and generated examinees in the second and third simulation studies. The averaged results were presented. The passing rate of the test is presumed to be 50%.

Model Settings

The indifference region $\delta$ for SPRT method is set as 0.2. The cut score for $\theta_0$ is 0. As a result, $\theta_1$ and $\theta_2$ are -0.2 and 0.2, respectively. Parameters $\alpha$ and $\beta$ for SPRT are set to 0.05. In addition, 18 integer values are generated for the step size $S$ in LA-CB methods, ranging from 3 to 20.

At the beginning of the test, the first three items are always selected randomly because we lack the knowledge to compute the Fisher information. Each following item is selected randomly from the two best items at the current step where the best item refers to the one with maximized priority index, Fisher information, or minimized weighted index, depending on which the method is used.

The Expected a posteriori (EAP) method with a prior of $\mathcal{N}(0,1)$ is used to estimate $\theta$ when all responses are 0s or 1s. If responses contain both values of 0 and 1, the maximum likelihood estimation (MLE) method is applied.

### 1.3.3 Evaluation Criteria

Various criteria are used to analyze and compare the two newly proposed methods with traditional methods. Results are evaluated based on the following four main aspects. Note that the last criterion is for the first simulation study only.

1. Classification accuracy: Three criteria are used for classification accuracy comparison in the simulations, including classification error rate (CER), Type I error rate (Type I ER) and Type II error rate (Type II ER). Meanwhile, mean square error (MSE) is also calculated as a measurement precision criterion, which is computed as:

$$MSE = \frac{\sum_{i=1}^{N}(\hat{\theta}_i - \theta_i)^2}{N}. \tag{1.31}$$

2. Content balancing: The total numbers of violated content constraints across various examinees are evaluated as a criterion for content balance. $V_i$ denotes the total number of constraints violated in all content areas of examinee $i$. The average number of constraint violated in a test is calculated by

$$\bar{V} = \frac{\sum_{i=1}^{N} V_i}{N}, \tag{1.32}$$

where $i$ denotes $i^{th}$ examinee and $N$ (i.e. 2,000) denotes the total number of examinees. The average value of $\bar{V}$ across different examinees, the maximum $\bar{V}$ and minimum $\bar{V}$ are given for comparison. The grand average $\bar{V}$ is defined as

$$\text{Average } \bar{V} = \frac{\sum_{p=P_0}^{P} \bar{V}_p}{P - P_0 + 1}, \tag{1.33}$$

where $\bar{V}_p$ denotes the average number of constraint violated of $p^{th}$ step size and $P_0$ and $P$ denote the minimum and maximum step sizes respectively we generated for LA-CB-C and LA-CB-A models. Maximum and minimum $\bar{V}$ are also calculated across different step sizes.

3. Exposure control: Four criteria are used for the purpose of evaluating exposure control across five different methods. They are the maximum item exposure rate, the proportion of overexposed items (items with exposure rate higher than 0.2), the proportion of unused items, and $\chi^2$. The $\chi^2$ is designated to measure the similarity between observed

and expected exposure rates ($ER$)

$$\chi^2 = \sum_{j=1}^{J} \frac{(ER_j - \bar{ER})^2}{\bar{ER}}, \tag{1.34}$$

where $j$ denotes $j^{th}$ item, $K$ denotes the total number of items, and $\bar{ER}$ shows the average exposure rate of all the items in the pool.

4. Test efficiency: To compare the test efficiency between two classification methods ACI and SPRT in the first study, the average test lengths $TL$s across various examinees are calculated conditioning on different step sizes. $\bar{TL}$ is expressed as:

$$\bar{TL} = \frac{\sum_{i=1}^{N} TL_i}{N}, \tag{1.35}$$

$TL_i$ stands for the test length received by $i^{th}$ examinee and $N$ (i.e. 2,000) is the total number of examinees.

### 1.3.4 Results of Simulation 1

The ACI and SPRT classification methods are adopted with LA-CB-C method and compared from the classification accuracy and test efficiency perspectives. The classification accuracy includes the classification error rate, Type I error rate, and Type II error rate. Therefore both the classification specificity and sensitivity can be shown. A reliable classification method is expected to provide both low classification error rate and short average test length.

The focus of the first study is to find out the most appropriate classification method which is a critical part of the VL-CCT design so that LA-CB methods can be further investigated on top of the recommended method. The classification method with better performance is applied in the following two studies.

Table 1.1 gives a comparison of classification accuracy and test efficiency between SPRT and ACI methods. The average $TL$ shows their performance on improving test efficiency with the benefit of variable length setting. While the average test length of SPRT is around 29.8, ACI has the average test length larger than 37.1. With SPRT as the termination criterion, the aver-

age test length is shortened by 19.7% compared to ACI. Figure 1.1 gives the comparison of total test lengths $TL$ between the two methods conditional on 18 step sizes. In addition, eighteen one-way ANOVA tests were run to compare the test lengths generated by the ACI and SPRT methods conditional on 18 step sizes. All p-values were reported to be smaller than $2 \times e^{-16}$ indicating the test lengths generated between the ACI and SPRT methods are significantly different.

Table 1.1: Overall Performance of Sequential Probability Ratio Test (SPRT) and Ability Confidence Interval (ACI) Classification Methods.

| Methods | SPRT | ACI |
|---|---|---|
| Avg. Test Length $TL$ | 29.85 | 37.11 |
| Grand Avg. Violated Constraints $\bar{V}$ | 0.011 | 0.017 |
| Average Classification Error Rate | 0.063 | 0.060 |
| Average Type I Error Rate | 0.033 | 0.030 |
| Average Type II Error Rate | 0.030 | 0.030 |

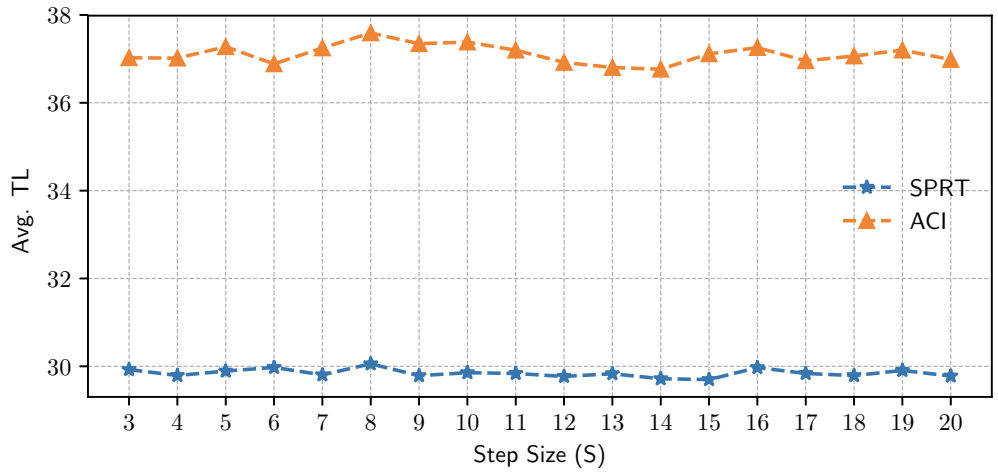


Figure 1.1: Average Test Length (TL) of Sequential Probability Ratio Test (SPRT) and Ability Confidence Interval (ACI).

The second row in Table 1.1 gives the average number of constraints violated in tests ($\bar{V}$). The value is 0.011 with SPRT and 0.017 with ACI. The result accords with the result given by $TL$ in Table 1.1 and Figure 1.1, since the ACI method tends to give a longer test so there is a higher probability of constraint violation.

The last part of the table presents the overall classification error rates of the two methods. The ACI method gives a slightly better performance with 6.0% error rate while SPRT has 6.3% error rate on classifying examinees. The difference of the average classification error rates between the ACI and SPRT methods is only 0.3%. Figure 1.2 presents the classification error rates of the two methods conditional on 18 step sizes, while Figure 1.3 and 1.4 give corresponding type I and type II error rates. The fluctuations of the curves are due to randomness from the test setting. Items are selected randomly from the two best ones and ability estimation errors also result in randomness in item selection procedure. In general, the differences of classification error rates between the two methods across 18 step sizes are quite small. The results show that comparative classification accuracy are achieved by the two methods.

Results given in Table 1.1, Figure 1.1, 1.2, 1.3 and 1.4 clearly show that SPRT improves test efficiency by shortening the test length by 19.7% without losing much capacity of maintaining high classification accuracy, which is 93.7% here. Similar conclusion that SPRT tends to gives a better performance in CCT can be found in other researches (Thompson, 2009; Babcock and Weiss, 2009; Lin, 2011) as well. Therefore, SPRT proves to be an efficient classification method which is used in simulation 2 and 3 for a further evaluation of the LA-CB methods.

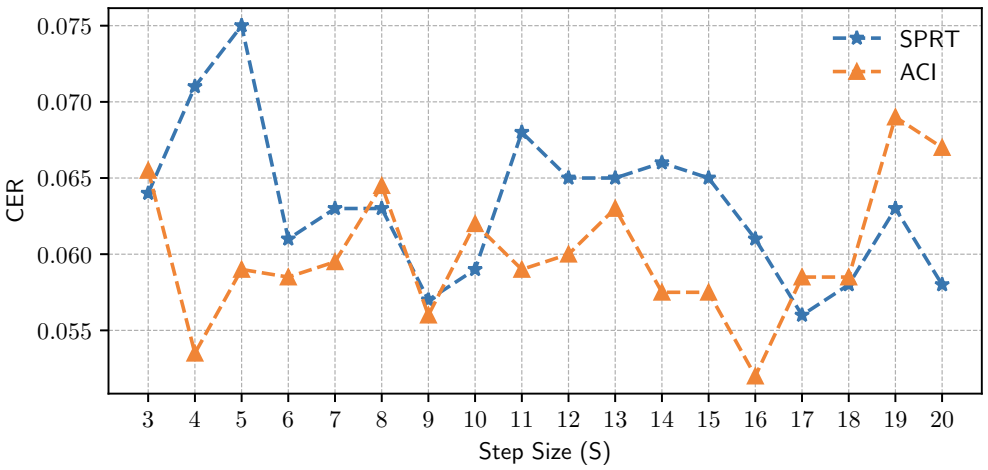


Figure 1.2: Classification Error Rate (CER) of Sequential Probability Ratio Test (SPRT) and Ability Confidence Interval (ACI).

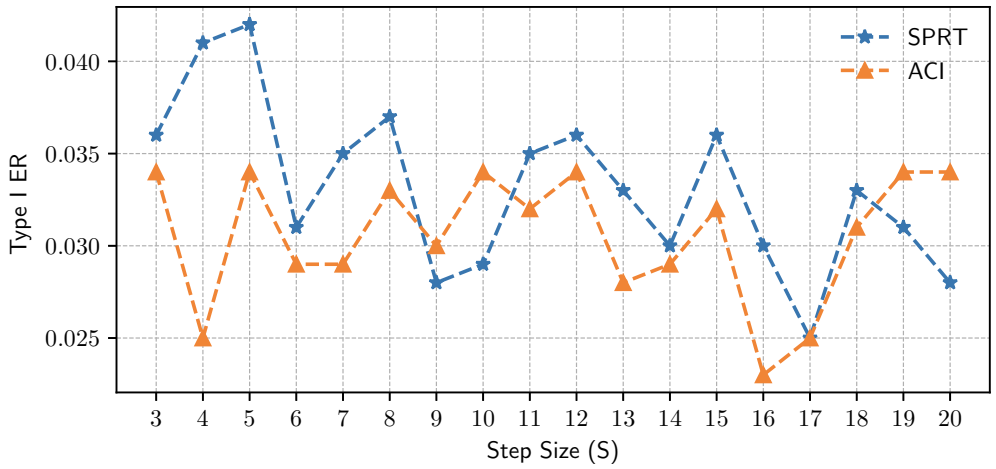


Figure 1.3: Type I Error Rate (ER) of Sequential Probability Ratio Test (SPRT) and Ability Confidence Interval (ACI).

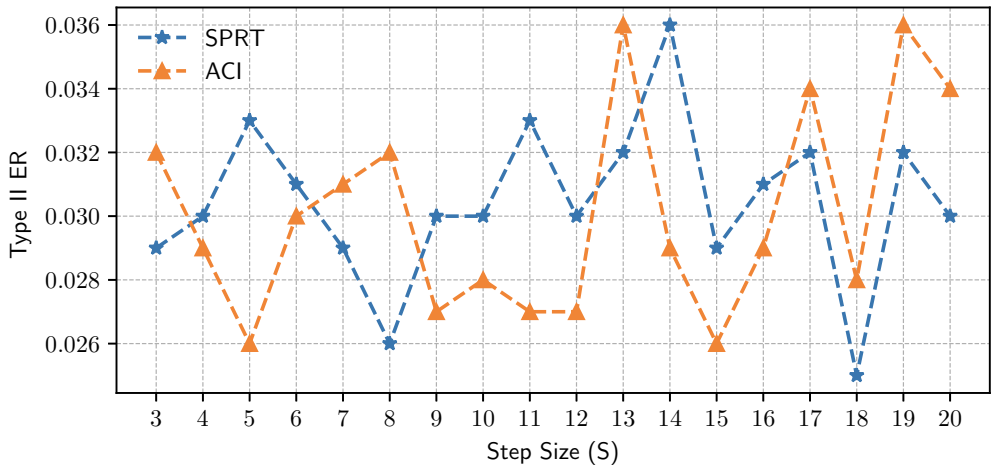


Figure 1.4: Type II Error Rate (ER) of Sequential Probability Ratio Test (SPRT) and Ability Confidence Interval (ACI).

### 1.3.5 Results of Simulation 2

Eighteen step sizes are generated for a comparison of LA-CB-C method with the CWI, MPI and the baselines of the MFI and randomized methods. The influence of different step sizes will be evaluated from different perspectives mentioned above. SPRT is adopted here as the classification method.

Since the LA-CB-C method is designed as a content balancing item selection method without sacrificing classification accuracy, criteria including

classification accuracy, content balancing and exposure control are recorded for a comparison. We replicated the simulation by 20 times and averaged the resulting values to give a reliable result.

Figure 1.5 and 1.6 present the classification error rates and the numbers of constraints violated across different step sizes with the LA-CB-C method. The trendline is a linear regression line which gives the linear trend of those two. Obviously, as the step size increases, the classification accuracy rate improves slightly, with error rate decreasing (see Figure 1.5). At the same time, the number of violated constraints increases greatly with larger step sizes (see Figure 1.6). There clearly exists a trade-off between classification accuracy and content balancing regarding different step sizes. With a decreasing step size, content constraints of selected items can be better controlled, with a slight loss of classification accuracy.

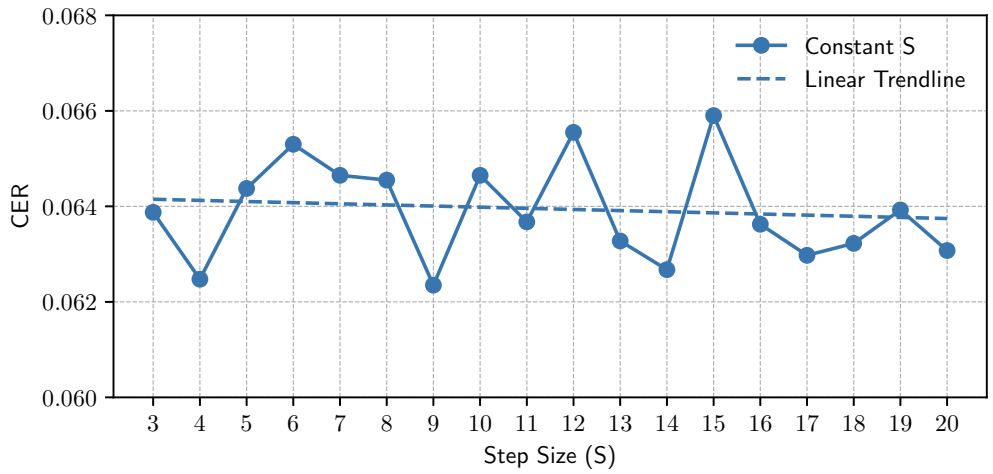


Figure 1.5: The LA-CB-C Method Classification Error Rate (CER) with Linear Trendline.

Table 1.2 and Figure 1.7 present the classification accuracy achieved by the LA-CB-C method under 18 step sizes and the MPI and CWI methods, compared to the baseline of MFI and randomized item selection methods. The results show that the randomized method has the highest CER, the MFI method has the lowest, while the CERs of the LA-CB-C, MPI and CWI methods lie in the middle. The average CER of the LA-CB-C method is 6.4%, slightly higher than the MPI method's 6.2% but much lower than the CWI method's 7.8%. There is no obvious optimum step size that achieves

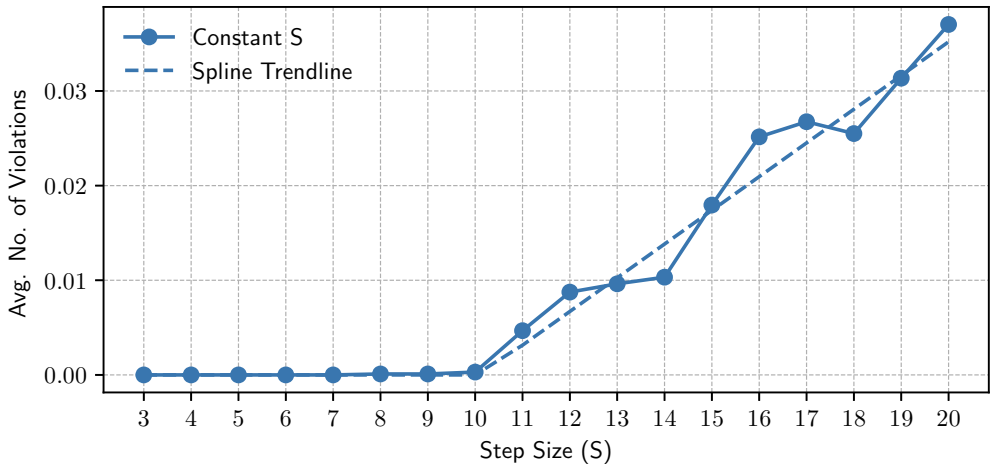


Figure 1.6: Average Number of Constraint Violations ($\bar{V}$) of the LA-CB-C Method with Spline Trendline.

lowest error rate. Classification error rates of the LA-CB-C method under different step sizes all lie within the range 6.2% and 6.6%.

Meanwhile, Figure 1.6 shows that the average number of constraints violated ($\bar{V}$) in the LA-CB-C method with different step sizes are all under 0.040, while $\bar{V}$ of the MPI method is 0.054, the MFI method is 8.14, the CWI method is 2.23 and the randomized method is 7.02 (see Table 1.5). Particularly, with step size from 3 to 10, the LA-CB-C method has no constraint violation. Even with step size 20, which has the highest $\bar{V}$, the LA-CB-C method still gives a better performance on constraint management than other four methods. This shows that the LA-CB-C method significantly improves content balancing compared to the MPI and CWI methods.

Table 1.3 shows that the exposure rate is well-controlled by both the LA-CB-C and MPI methods, especially compared with the MFI method. The maximum exposure rates of the LA-CB-C and MPI methods are both under 0.2, and all items in the pool are used.

The results show that the LA-CB-C method has comparable classification accuracy with the MPI and MFI methods, better than the CWI method. In addition, the LA-CB-C method generates much smaller number of violated constraints than the other four methods, while having similar the exposure control performance with the MPI method but better than the CWI, MFI and randomized methods. The results indicate that from the content constraint

Table 1.2: Classification Error Rates (ER) and Mean Square Error (MSE) of the LA-CB-C Method and Three Other Methods under 18 Step Sizes $S$.

| $S$ | CER | Type I ER | Type II ER | MSE |
|---|---|---|---|---|
| 3 | 0.064 | 0.034 | 0.030 | 0.074 |
| 4 | 0.062 | 0.033 | 0.030 | 0.074 |
| 5 | 0.064 | 0.033 | 0.031 | 0.075 |
| 6 | 0.065 | 0.034 | 0.031 | 0.075 |
| 7 | 0.065 | 0.033 | 0.031 | 0.075 |
| 8 | 0.065 | 0.033 | 0.032 | 0.074 |
| 9 | 0.062 | 0.033 | 0.030 | 0.075 |
| 10 | 0.065 | 0.034 | 0.031 | 0.076 |
| 11 | 0.064 | 0.032 | 0.032 | 0.075 |
| 12 | 0.066 | 0.034 | 0.031 | 0.075 |
| 13 | 0.063 | 0.033 | 0.030 | 0.075 |
| 14 | 0.063 | 0.033 | 0.030 | 0.075 |
| 15 | 0.066 | 0.034 | 0.032 | 0.075 |
| 16 | 0.064 | 0.033 | 0.031 | 0.074 |
| 17 | 0.063 | 0.033 | 0.030 | 0.075 |
| 18 | 0.063 | 0.033 | 0.030 | 0.075 |
| 19 | 0.064 | 0.032 | 0.032 | 0.075 |
| 20 | 0.063 | 0.033 | 0.030 | 0.075 |
| LA-CB-C Average | 0.064 | 0.033 | 0.031 | 0.075 |
| Maximum Priority | 0.062 | 0.032 | 0.030 | 0.074 |
| Content Weighted | 0.078 | 0.040 | 0.039 | 0.118 |
| Maximum Information | 0.054 | 0.032 | 0.022 | 0.054 |
| Randomized | 0.084 | 0.042 | 0.042 | 0.254 |

Table 1.3: Overall Exposure Control indices.

| Methods | LA-CB-C | Maximum Priority | Content Weighted | Maximum Information | Randomized |
|---|---|---|---|---|---|
| Max. exposure rate | 0.178 | 0.175 | 0.166 | 0.532 | 0.100 |
| Over exposed (%) | 0 | 0 | 0 | 3.2 | 0 |
| Never exposed (%) | 0 | 0 | 0 | 0 | 0 |
| $\chi^2$ | 20.297 | 20.228 | 4.009 | 83.041 | 0.153 |

management perspective, the LA-CB-C method outperforms all the other methods.

### 1.3.6 Results of Simulation 3

The LA-CB-A method is designed to improve the LA-CB-C method in terms of meeting the content constraints. The criteria to compare the LA-CB-A and LA-CB-C methods include classification accuracy, content balancing, and exposure control. The simulation was replicated by 20 times and the results were summarized in Table 1.4, 1.5, Figure 1.7 and 1.8.

The adaptive step size is used in the LA-CB-A method, expected to better control the content area constraints based on the LA-CB-C method. Table 1.4 and Figure 1.7 give the performance of the LA-CB-A method on classification accuracy compared to other methods. The average CER of the LA-CB-A the is 6.4%, same as that of the LA-CB-C method, close to the MPI method, slightly higher than the MFI method, and much lower than the CWI and randomized methods.

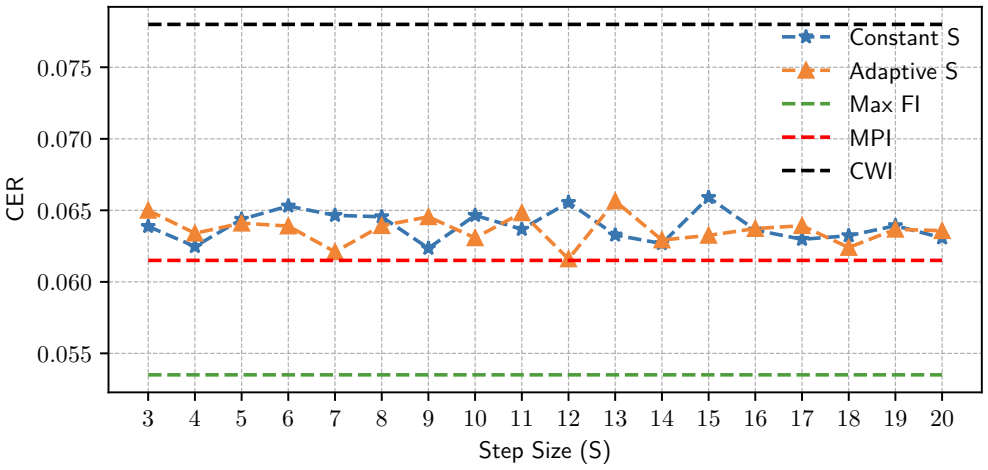


Figure 1.7: Classification Error Rate (CER) of the LA-CB methods with constant (Constant $S$) and adaptive (Adaptive $S$) step sizes and Maximum Priority (MPI), Maximum Information (Max FI) and Content Weighted (CWI) Methods.

Table 1.5 and Figure 1.8 present the overall content balancing performance achieved by the LA-CB-C and LA-CB-A methods, compared to the other four methods. Obviously, the LA-CB-C and LA-CB-A methods both outperform other methods, while the LA-CB-A method has the smallest average $\bar{V}$ (see Table 1.5). By deep-diving in the two methods under different step sizes, the LA-CB-A method controls $\bar{V}$ better than the LA-CB-C method

Table 1.4: LA-CB-A Classification Error Rates (ER) and Mean Square Error (MSE) under Different Step Sizes $S$.

| Constant $S$ | CER | Type I ER | Type II ER | MSE |
|---|---|---|---|---|
| 3 | 0.065 | 0.034 | 0.031 | 0.076 |
| 4 | 0.063 | 0.032 | 0.031 | 0.076 |
| 5 | 0.064 | 0.034 | 0.030 | 0.076 |
| 6 | 0.064 | 0.033 | 0.031 | 0.075 |
| 7 | 0.062 | 0.032 | 0.030 | 0.075 |
| 8 | 0.064 | 0.033 | 0.031 | 0.074 |
| 9 | 0.065 | 0.033 | 0.031 | 0.076 |
| 10 | 0.063 | 0.033 | 0.031 | 0.073 |
| 11 | 0.065 | 0.033 | 0.031 | 0.074 |
| 12 | 0.062 | 0.032 | 0.030 | 0.074 |
| 13 | 0.066 | 0.034 | 0.032 | 0.075 |
| 14 | 0.063 | 0.034 | 0.029 | 0.074 |
| 15 | 0.063 | 0.032 | 0.031 | 0.076 |
| 16 | 0.064 | 0.033 | 0.031 | 0.074 |
| 17 | 0.064 | 0.033 | 0.031 | 0.074 |
| 18 | 0.062 | 0.032 | 0.030 | 0.075 |
| 19 | 0.064 | 0.033 | 0.030 | 0.074 |
| 20 | 0.064 | 0.033 | 0.031 | 0.075 |
| LA-CB-A Average | 0.064 | 0.033 | 0.031 | 0.075 |
| Maximum Priority | 0.062 | 0.032 | 0.030 | 0.074 |
| Content Weighted | 0.078 | 0.040 | 0.039 | 0.118 |
| Maximum Information | 0.054 | 0.032 | 0.022 | 0.054 |
| Randomized | 0.084 | 0.042 | 0.042 | 0.254 |

(with smaller $\bar{V}$) especially when the step size gets larger. It makes sense since more constraints tend to be violated when the step size gets larger, while the LA-CB-A method gives a look-ahead prediction of the test length with an adaptive step size, which is no larger than the constant step size in LA-CB-C. It is also worth noting that both the LA-CB-C and LA-CB-A methods control constraints almost perfectly when the step size is smaller than 11 (Figure 1.8). Table 1.6 presents the LA-CB-A method also controls exposure rate very well and is comparable with the LA-CB-C method.

The results show that the LA-CB-A method does manage constraints better than the LA-CB-C method with high classification accuracy and low exposure rate. The LA-CB-A method improves the performance of control-

ling content constraints significantly especially under the condition of larger step sizes and gives a perfect constraint management when the step sizes is small.

Table 1.5: Summary of Content Constraint Violations ($\bar{V}$).

| Measures | Average $\bar{V}$ | Max $\bar{V}$ | Min $\bar{V}$ |
|---|---|---|---|
| LA-CB-C | 0.0110 | 0.0370 | 0 |
| LA-CB-A | 0.0102 | 0.0319 | 0 |
| Maximum Priority* | 0.0540 | - | - |
| Content Weighted* | 2.2295 | - | - |
| Maximum Information* | 8.1380 | - | - |
| Randomized* | 7.0230 | - | - |

Note: The table summarizes the statistics of $\bar{V}$ across 18 step sizes. Methods with (*) do not include step sizes to make item selections and therefore maximum or minimun $\bar{V}$ are not applicable.

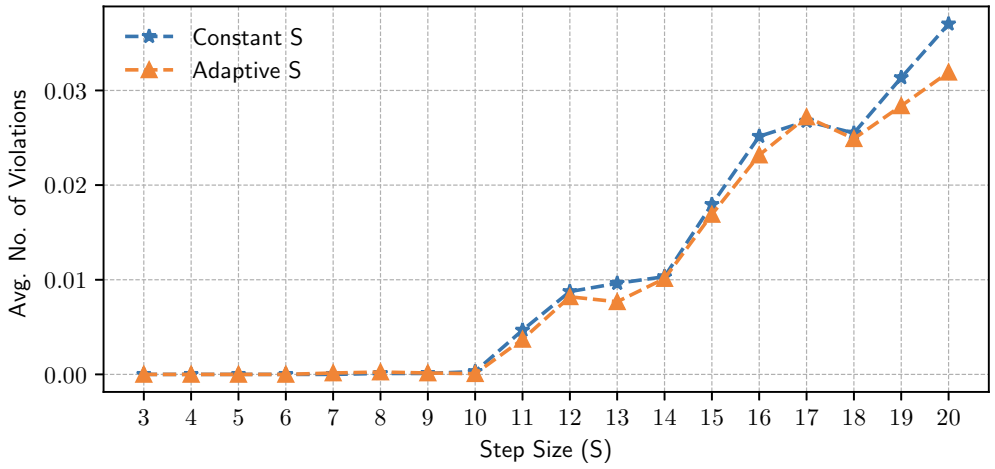


Figure 1.8: Average Number of Content Constraint Violations ($\bar{V}$) of the LA-CB methods with constant (Constant $S$) and adaptive (Adaptive $S$) step sizes.

## 1.4 DISCUSSION

The results reported in the preceding section indicate that the proposed LA-CB methods with SPRT are promising solutions to content constrained VL-CCT tests. First, the results show that the LA-CB methods perform better than the CWI and MPI methods in controlling constraints (e.g., content

Table 1.6: Overall exposure control indices.

| Methods | LA-CB-C | LA-CB-A |
|---|---|---|
| Maximum exposure rate | 0.178 | 0.178 |
| Over exposed (%) | 0 | 0 |
| Never exposed (%) | 0 | 0 |
| $\chi^2$ | 20.297 | 20.289 |

area constraints and exposure rate), while still maintaining high classification accuracy. Second, with adaptive step sizes, the trade-off between the classification accuracy and constraint management can be alleviated. Specifically, the LA-CB methods reduce the number of constraint violations without sacrificing classification accuracy. Third, both the LA-CB-C and LA-CB-A methods are flexible and easy to implement in practice.

The VL-CCT program shows its advantages in improving test efficiency and accuracy. Yet, due to the lack of information on test length, the non-statistical constraints are hard to control which is very different from the fixed-length CCT program. Now with the proposed LA-CB methods, it is possible to control content constraints and achieve high classification accuracy simultaneously. As such, the VL-CCT approach can play a more important role in future large-scale tests.

# CHAPTER 2

# OPTIMAL HIERARCHICAL LEARNING PATH DESIGN WITH REINFORCEMENT LEARNING

## 2.1 INTRODUCTION

In a traditional classroom, a teacher uses the same learning material (e.g. textbook, instruction pace, etc.) for all students. However, the selected material may be too hard for some students and too easy for some other students. Further, some students may take longer time to learn than others. Such a learning process may not be efficient. These issues can be solved if the teacher can make an individualized learning plan/designs an individualized learning path for each individual student: Select an appropriate learning material according to each student' ability and let a student learn at her/his own pace. Considering that a very low teacher-student ratio is required, such an individualized adaptive learning plan may be too expensive to be applied to all students. As such, adaptive learning systems are developed to provide individualized adaptive learning for all students/learners. In particular, with the fast growth of digital platforms, globally integrated resources, and machine learning algorithms, the adaptive learning systems are becoming increasingly more affordable, applicable, and efficient (Zhang and Chang, 2016).

An adaptive learning system—also referred to as a personalized/ individualized learning or intelligent tutoring system—aims to provide a learner with optimal and individualized learning experience or instructional materials so that the learner can master prespecified skills/reach a certain achievement level in a shortest time or reach as high as possible an achievement level in a fixed period of time. First, learners' historical data are used to estimate his/her evolving progress on selected skills. Then, according to the number of skills she/he has mastered or the level of her/his proficiency, the system finds the optimal learning strategy/plan (called *policy* in the rest of the chapters) which selects the most appropriate learning material for the learner. After the learner finishes the learning material, an assessment is given to the learner and her/his skill status or proficiency level is updated and is used by the adaptive learning system to choose the next most appropriate learning

material for the learner. Such process is repeated until the learner masters all prespecified skills or achieves a certain proficiency level.

Several studies have provided innovative approaches to adaptive learning systems. One of the directions with respect to adaptive learning systems is to track learners' skill acquisition and model changes in their skills, referred as learning paths. For example, cognitive diagnosis models (CDMs), known as the foundation of assessing learners' mastery of skills, are extended to model their learning paths (Chen et al., 2018b; Wang et al., 2018). The knowledge tracing method (Corbett and Anderson, 1994) functions similarly in modeling learning but focusing on one skill each time (Studer, 2012). The modeled individualized learning trajectories can be used as priors to provide learning recommendations for learners in the future. Another direction towards personalized learning is to find optimal learning path that recommends learning materials for adaptive learning systems (Chen et al., 2018c; Tang et al., 2019) or intelligence tutoring systems (Brusilovsky and Peylo, 2003; Lan and Baraniuk, 2016). Personalized learning materials are thus selected by the systems for learners and each individual's total learning time to master skills is shortened.

However, there are two challenges in the existing approaches. First, aforementioned research studies except Tang et al. (2019) typically characterize the learning path as a Markov decision process (MDP) assuming its transition probabilities are known. However, the transition probabilities are hardly known in practice. As a matter of fact, the transition paths of learners' skills/attributes are unobservable and may vary across different learning materials. Second, skills are typically assumed to be unstructured without considering skill hierarchical structure and mastery levels related to the skills in previous learning tracking research. Ignoring skill hierarchy and mastery levels may contaminate classification results (Tu et al., 2018).

Tang et al. (2019) applied the model-free Q-learning method with linear function approximation to solve some MDP problems as showcases without the known transition probability assumption. However, their showcases are relatively trivial without considering the intrinsic hierarchical structures among skills. The optimal learning policies in the showcases can be found by subject experts directly based on contents of learning materials and the skill hierarchy structure that can be derived from empirical considerations or curriculum development (Leighton et al., 2004). Besides, it is well known in

reinforcement learning communities that the Q-learning method with linear function approximation may fail to find an optimal policy in certain situations (Geramifard et al., 2013; Melo and Ribeiro, 2007; Thrun and Schwartz, 1993).

In this chapter, we address those challenges by proposing an integrated adaptive learning system equipped with the optimal learning policy, an algorithm that designs the optimal learning paths for learners, without the known transition probability assumption and a data-driven method that takes the skill hierarchy into consideration. After finding the optimal learning policy and selecting the most appropriate learning materials for new learners, the algorithm keeps the system being trained using the new learners' information (data). In the rest of the chapter, we refer to a set of skills as *attribute profile* and a skill as an *attribute* which are conventional terms in CDMs. Specifically, we first develop a unified hierarchical learning model to explicitly characterize attribute hierarchy and mastery levels of attributes, or called ordered polytomous attributes in CDM literatures (Chen and de la Torre, 2013; Karelitz, 2004; Templin, 2004), which, albeit important, have not been addressed yet in existing adaptive learning system research. Attribute hierarchy widely exists in practice as the curriculum is designed to follow a hierarchical structure. For example, mathematics contains many attributes which are related and often constructed on one another (Sternberg and Ben-Zeev, 1996). In addition, the optimal learning path cannot be easily designed by subject experts due to unobservable transition models of mastery levels across hierarchical attributes. Thus, it is crucial to consider hierarchical attributes and their mastery levels when building an adaptive learning system. We model the mastery levels related to hierarchical attributes following the same form of CDMs for binary attributes, instead of using polytomous attributes which can be accommodated by only a few CDMs (Chen and de la Torre, 2013); therefore, the latent attributes and their mastery levels can be first pre-specified as binary labels by subject experts and later estimated using conventional CDMs. The optimal learning path is next formulated as an MDP, in which the state is the (discrete) attribute profile of a learner, the action is the (discrete) learning material selected to the learner. The proposed hierarchical learning model transforms the polytomous attributes to take a uniform form which is flexible and easy to implement in adaptive learning systems and can accommodate various types of attribute hierarchies

(Leighton et al., 2004) which will be discussed in the later section. In addition, the number of latent states to be estimated is largely reduced with regards to the restricted state space defined in the model.

Second, a data-driven reinforcement learning (RL) method is applied to finding the optimal learning policy. Reinforcement learning is a type of machine learning technique that takes suitable action so as to achieve the objective (e.g., minimize total learning time) by interacting with the environment through trial-and-error search and collecting feedback. Using RL techniques, the proposed adaptive learning system is fully data-driven and does not require prior information to solve the MDP. After each stage of learning, a set of items are distributed to the learners, whose responses to these items are collected by the adaptive learning system. Learners' latent attributes are estimated and their attribute profile status is updated based on the responses using CDMs. We compare the data-driven RL method with a heuristic method that randomly selects a material among all available ones via simulation studies constructed under the proposed hierarchical learning model. The results indicate that the data-driven RL method can quickly find a learning policy outperforming the heuristic one.

## 2.2 PRELIMINARIES

In this section, we provide some background on CDMs and the Markov model which will be used to characterize the learning paths.

### 2.2.1 Cognitive Diagnosis Models

CDMs are psychometric models that examine learners' mastery of specific attributes at a fine-grained level. Attributes in CDMs are assumed to be latent and discrete. The element of an attribute profile takes binary values if only the mastery or non-mastery of an attribute is modeled. Latent attribute profiles can be reflected by responses given by examinees to items measuring one or more attributes. They are ideal frameworks that aid in identifying optimal learning materials to be distributed next since they keep track of learners' different attributes considering their multidimensional features. These models provide a summary information in the form of attribute profiles, the element of which represents the mastery level related to an attribute by examinees.

Most CDMs require the construction of a Q-matrix (Embretson, 1984) for implementation. To be specific, suppose that the adaptive learning system considers $N$ attributes and contains $J$ items. The Q-matrix is a $J \times N$ matrix whose element $q_{jn}$, $j = 1, \cdots, J$, $n = 1, \cdots, N$, on the $j^{\text{th}}$ row (item) and $n^{\text{th}}$ column (attribute) taking binary values, indicates whether the $j^{\text{th}}$ item is associated with the $n^{\text{th}}$ attribute. The Q-matrix specifies the cognitive specification for each test item explicitly (de la Torre, 2009).

An example is provided to illustrate the construction of Q-matrix. Consider the mixed attributes in the system including addition and multiplication. The item "$5 + 4$" requires addition attribute for itself to be answered correctly, while "$5 + 2 \times 2$" measures both addition and multiplication attributes. Thus the corresponding row of the Q-matrix for the first item is $(1, 0)$ and that for the second is $(1, 1)$. The Q-matrix provides a method to formulate the conditional independence between item responses and attribute profiles. That is, conditioning on measured attributes, item responses are independent of irrelevant attributes. The Q-matrix is generally specified before a test and further improved based on learners' responses during the test (Chen et al., 2018a; Liu et al., 2012).

One widely-used CDM is the deterministic inputs, noisy "and" gate (DINA) model (Junker and Sijtsma, 2001) which is both tractable and interpretable. In the DINA model, the probability of correctly answering an item is defined based on the Q-matrix. Following the same notation as above, assume $N$ attributes and $J$ items in the adaptive learning system. Let $\boldsymbol{\alpha_i}$ be the attribute profile for the $i^{\text{th}}$ learner, where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{iN})$ and each element of $\boldsymbol{\alpha}_i$ belongs to $\{0, 1\}$. A value of 1 indicates a mastered attribute and 0 indicates an unmastered attribute. Let $X_{ij}$ be the response of learner $i$ to item $j$, $j = 1, \cdots, J$, where $X_{ij} = 1$ indicates a correct answer while 0 indicates an incorrect one. Therefore, the probability of a correct answer conditional on the attribute profile is defined as

$$\mathbb{P}(X_{ij} = 1 | \boldsymbol{\alpha}_i) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}, \tag{2.1}$$

where $\mathbb{P}$ denotes probability, $\eta_{ij}$ indicates whether or not the learner $i$ has mastered all attributes required for the item $j$. The value of $\eta_{ij}$ is 1 if the learner possesses all attributes and is 0 if the learner lacks at least one of the

required attributes. Mathematically, the value $\eta_{ij}$ is defined as

$$\eta_{ij} = \prod_{n=1}^{N} \alpha_{in}^{q_{jn}}.$$

In the model, the value $q_{jn}$ is an entry in the Q-matrix, $s_j$ denotes the slipping parameter—the probability of a learner possessing all attributes required in item $j$ yet failing to answer correctly, e.g.,

$$s_j = \mathbb{P}(X_{ij} = 0 | \eta_{ij} = 1),$$

and $g_j$ denotes the guessing parameter—the probability of correctly answering the item without required attributes, e.g.,

$$g_j = \mathbb{P}(X_{ij} = 1 | \eta_{ij} = 0).$$

CDMs are classified as non-compensatory and compensatory models (DiBello et al., 2007). The DINA model is a non-compensatory model since it assumes the learner who lacks any of the required attributes will fail to answer the item correctly unless guessing. Other non-compensatory models include noisy input, deterministic, "and" gate (NIDA) model (Maris, 1999), the reparameterized unified model (RUM) or fusion model (Hartz, 2002), and the reduced reparameterized unified model (r-RUM; Roussos et al., 2007). Unlike non-compensatory models, compensatory models allow a high ability attribute to compensate for a low ability attribute on another dimension. Compensatory models include deterministic input noisy "or" gate (DINO) model (Templin and Henson, 2006) and additive cognitive diagnostic model (ACDM; de la Torre, 2011). More general CDMs have been developed to include many non-compensatory and compensatory models (Henson et al., 2009; de la Torre, 2011). Both non-compensatory and compensatory models are well-examined in modeling diagnostic attributes and the estimation of CDMs (e.g., expectation-maximization and Markov Chain Monte Carlo (MCMC) algorithms) as well as their software programs (Bolt et al., 2008; de la Torre, 2009; George et al., 2016; Muthén and Muthén, 1998; Templin et al., 2010; von Davier, 2006).

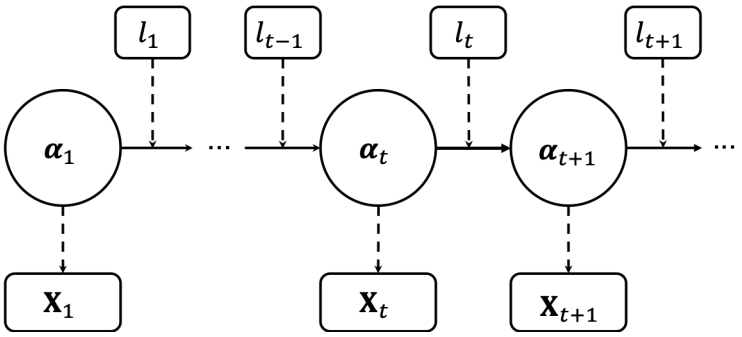


Figure 2.1: An illustration of learning path with the Hidden Markov Model ($\boldsymbol{\alpha_t}$ denotes the latent attribute profile, $l_t$ denotes the learning material and $X_t$ denotes the learner's responses at time step $t$).

### 2.2.2 Learning Paths with the Hidden Markov Model

Learning paths can be modeled by the HMM because the attribute profiles are latent (Kaya and Leite, 2017; Li et al., 2016; Norris, 1998; Wang et al., 2018). The Markov model specifies that a learner's next state, after provided with a certain learning material, will only depend on his or her current state and the material. Figure 2.1 illustrates how to model the learning path with an HMM. Define the attribute profile as the state in the Markov model, denoted as $\boldsymbol{\alpha}_t = (\alpha_{1,t}, \alpha_{2,t}, \cdots, \alpha_{N,t})$ for a learner with $N$ attributes at time step $t$. The state transition is as follows:

$$\boldsymbol{\alpha}_t \times l_t \rightarrow \boldsymbol{\alpha}_{t+1}, \tag{2.2}$$

where $l_t$ denotes the learning material distributed at time $t$, and $l_t \in \mathcal{L} = \{l_1, \cdots, l_L\}$, which is the set of all learning materials. The transition process from current state to the next is thus formulated as a Markov decision process (MDP).

The learning paths with latent attribute profiles can either be considered as a partially observable MDP (Kaelbling et al., 1998), or two separate components—one with a psychometric model and one MDP. In both cases, we assume no retrogression exists—once learners master an attribute, they will not lose it, e.g., for $\forall n \in \{1, \cdots, N\}$

$$\mathbb{P}(\alpha_{n,t+1} = 1 | \alpha_{n,t} = 1) = 1, \tag{2.3}$$

and

$$\mathbb{P}(\alpha_{n,t+1} = 0 | \alpha_{n,t} = 1) = 0. \tag{2.4}$$

In this study, the CDM and an HMM are used to estimate the attribute profiles. Specifically, given time-invariant item parameters and a proper psychometric model such as a CDM, the attribute profile $\boldsymbol{\alpha}_t$ of a learner at time step $t$ can be estimated from item responses at time step $t$, denoted as $\mathbf{X}_t$, as shown in Figure 2.1. Take the DINA model as an example. Given item responses, the attribute profile can be estimated through (2.1).

## 2.3 MODELS AND ALGORITHMS

In this section, a uniform and flexible hierarchical learning model that considers attribute hierarchy and mastery levels of attributes is constructed in the framework of CDMs. The problem of finding the optimal learning policy is next formulated as an MDP and solved by a RL algorithm, an efficient and stable algorithm for solving MDPs with unknown models.

### 2.3.1 Hierarchical Learning Model

In an adaptive learning system, learners are first given a test to assess what mastery levels they have reached on different attributes, and next provided with learning materials based on their responses so as to improve mastery levels within shortest learning steps under the optimal learning policy. In the first step, the psychometric model such as a CDM is applied to estimate the latent attribute profiles. In the second step, a material is selected for learners by the optimal learning policy accordingly based on their estimated attribute profiles. To characterize the transition process of learners' hierarchical attributes considering mastery levels, the unified and flexible hierarchical learning model is proposed to incorporate both the hierarchical structure and mastery levels in presenting attribute profiles with binary values, such that attribute profiles can be easily estimated using conventional CDMs. In addition, assumptions regarding the transition of mastery levels in hierarchical attributes which conform to a realistic learning process are proposed in the model.

Attribute hierarchy method (AHM) was first proposed to deal with situations where cognitive attributes are hierarchically related and thus correlated

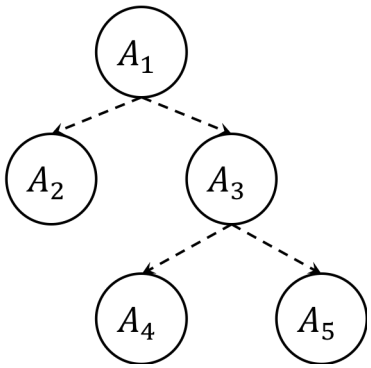


Figure 2.2: A divergent hierarchical structure among five cognitive attributes.

with each other (Leighton et al., 2004). In particular, the AHM investigates precedence ordering of cognitive competencies required to solve test problems. It has four different structures including linear, convergent, divergent and unstructured. An intuitive example of the hierarchical structure is how learners learn addition "+" and multiplication "×". Addition is considered as a prerequisite for multiplication. Learners are able to learn multiplication only after they fully understand addition or at least are equipped with basic knowledge of it.

All structures investigated by AHM can be split into dependent relationships between two attributes. For example, Figure 2.2 exhibits the divergent structure among 5 attributes, denoted as $A_n$, $n = 1, \cdots, 5$. The hierarchical structure among the five can be split to the four dependent links shown as dotted arrow line in Figure 2.2. That is, $A_1$ is a prerequisite of $A_2$ and $A_3$, while $A_3$ is a prerequisite of $A_4$ and $A_5$. Therefore, the hierarchical learning model is proposed to capture the relationship between two dependent attributes and the complete hierarchical attribute structure can be expressed based on paired attributes accordingly.

First, assumptions on the link between two dependent attributes are constructed. Assume attribute $A_1$ is prerequisite to attribute $A_2$. There are $K$ different mastery levels for each attribute. Denote the lack of attribute $A_n$ as $A_n^{(0)}$, $n \in \{1, 2\}$, and $K$ different mastery levels as $A_n^{(1)}, \cdots, A_n^{(K)}$. Whether or not possessing a certain level of each attribute is binary. Throughout this chapter, we make the following assumptions:

**A**2.1. Learners can only possess a high mastery level after they have mastered

the lower level of the same attribute. That is,

$$\mathbb{P}(A_n^{(k)} = 1 | A_n^{(k-1)} = 0) = 0, k \in \{2, \cdots, K\}. \tag{2.5}$$

**A**2.2. Certain mastery level of attribute $A_2$ can only be learned after the same mastery level of attribute $A_1$ is achieved. That is,

$$\mathbb{P}(A_2^{(k)} = 1 | A_1^{(k)} = 0) = 0, k \in \{1, \cdots, K\}. \tag{2.6}$$

**A**2.3. The probability of a learner to master a certain mastery level of the attribute $A_{n_1}$ conditional on mastering a higher level of attribute $A_{n_2}$ is no smaller than mastering a lower level of attribute $A_{n_2}$, $\{n_1, n_2\} = \{1, 2\}$. That is, for $k \in \{1, \cdots, K-1\}$ and $\tilde{k} \in \{k, \cdots, K-1\}$,

$$\mathbb{P}(A_2^{(k)} = 1 | A_1^{(\tilde{k}+1)} = 1) \geq P(A_2^{(k)} = 1 | A_1^{(\tilde{k})} = 1), \tag{2.7}$$

and for $k \in \{2, \cdots, K\}$ and $\tilde{k} \in \{1, \cdots, k-1\}$,

$$\mathbb{P}(A_1^{(k)} = 1 | A_2^{(\tilde{k})} = 1) \geq P(A_1^{(k)} = 1 | A_2^{(\tilde{k})} = 0). \tag{2.8}$$

We next model different mastery levels of hierarchical attributes to be elements of attribute profiles as in CDMs. Assume that an attribute $A$ has $K$ different mastery levels, denoted as $A^{(1)}, A^{(2)}, \cdots, A^{(K)}$. If a learner reaches level $k$, $\forall k \in \{1, \cdots, K\}$ on this attribute, we have $A^{(k)} = 1$, while $A^{(\kappa)} = 1$, $\forall \kappa < k$ and $A^{(\kappa')} = 0$, $\forall \kappa' > k$. The attribute profile $\boldsymbol{\alpha}$ with two attributes $A_1$ and $A_2$, each with $K_1$ and $K_2$ mastery levels respectively is thus represented by $\boldsymbol{\alpha} = (A_1^{(1)}, \cdots, A_1^{(K_1)}, A_2^{(1)}, \cdots, A_2^{(K_2)})$. Note that $K_1$ is not related to $K_2$, that is, they can be equal or different. An example of a Q-matrix for two hierarchical attributes with two mastery levels is provided in Table 2.1. In this example, attribute addition (+) is presumed to be a prerequisite of attribute multiplication (×). One-digit calculation is assumed to be the low mastery level while two-digit calculation is assumed to be the high mastery level for both operations.

The three assumptions can be further explained intuitively using the same example. Under Assumption **A**2.1, a high level of any attribute, e.g., $+^{(2)}$, cannot be possessed until its lower level $+^{(1)}$ is mastered. Furthermore,

Table 2.1: A Q-matrix of Addition (+) and Multiplication (×) Attributes with Two Levels.

| Item | $+^{(1)}$ | $+^{(2)}$ | $\times^{(1)}$ | $\times^{(2)}$ |
|---|---|---|---|---|
| $7+2$ | 1 | 0 | 0 | 0 |
| $11+4*5$ | 1 | 1 | 1 | 0 |
| $12*31$ | 1 | 1 | 1 | 1 |

Table 2.2: State Space for Addition (+) and Multiplication (×) Attributes with Two Levels.

| State | $+^{(1)}$ | $+^{(2)}$ | $\times^{(1)}$ | $\times^{(2)}$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 1 | 0 |
| 5 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 |

under Assumption **A**2.2, if the learner has not reached certain level on the prerequisite attribute, e.g., $+^{(2)} = 0$, the higher level of $\times$ cannot be reached such that $\mathbb{P}(\times^{(2)} = 1|+^{(2)} = 0) = 0$. Equation (2.7) in Assumption **A**2.3 indicates the ability of multiplication is likely easier to learn when the ability of addition reaches a higher level, e.g., $\mathbb{P}(\times^{(2)} = 1|+^{(2)} = 1) \geq P(\times^{(2)} = 1|+^{(1)} = 1)$, while equation (2.8) in Assumption **A**2.3 indicates the ability of addition is likely easier to learn when the ability of multiplication reaches a higher level, e.g., $\mathbb{P}(+^{(2)} = 1|\times^{(1)} = 1) \geq P(+^{(2)} = 1|\times^{(1)} = 0)$.

To incorporate the attribute hierarchy, the state space is constructed following the hierarchical learning model assumptions. Without a hierarchical attribute structure, $2^4 = 16$ states shall be included in the HMM with respect to 4 attributes. With the hierarchical learning model, the state space is reduced to 6 states shown as rows in Table 2.2. As a result, the attribute profile of a learner at time step $t$, that is, $\boldsymbol{\alpha}_t$, can only be one of the rows in Table 2.2.

All attribute hierarchy can be generalized by the hierarchical learning model. More strict assumptions can be added if necessary, in practice. For example, an attribute cannot be learned before its prerequisite is fully mastered. If so, the state space of the example in the simulation study will be further restricted by removing row 4 in Table 2.2.

The design of hierarchical learning model makes it possible to incorporate not only attribute hierarchy, but also different mastery levels of attributes in CDMs. The model follows the common form of CDMs so that the restricted Q-matrix is easy to construct, and parameters in CDMs as well as attributes can be estimated easily (Tu et al., 2018). In addition, the hierarchical design largely reduces the number of attributes to be estimated in CDMs.

### 2.3.2 Markov Decision Process and Reinforcement learning

#### Primer on Markov Decision Process

Before presenting the formulation for the problem of finding the optimal learning policy, we first briefly review MDPs. An MDP is characterized by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is a Markovian transition model, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a reward function, and $\gamma \in [0, 1)$ is a discount factor (Sutton and Barto, 2018). A transition sample is defined as $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')$, where $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$, $r = \mathcal{R}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$ is a scalar reward when the state transitions into state $\boldsymbol{s}'$ from state $\boldsymbol{s}$ after taking action $\boldsymbol{a}$.

Let $\boldsymbol{S}_t$ and $\boldsymbol{A}_t$ denote the state and action at time step $t$, respectively, and $R_t$ denote the reward obtained after taking action $\boldsymbol{A}_t$ at state $\boldsymbol{S}_t$. Note that $\boldsymbol{S}_t$, $\boldsymbol{A}_t$, and $R_t$ are random variables. When both $\mathcal{S}$ and $\mathcal{A}$ are finite, the transition model $\mathcal{P}$ can be represented by conditional probability, that is,

$$\mathcal{P}_t(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) = \mathbb{P}(\boldsymbol{S}_{t+1} = \boldsymbol{s}'|\boldsymbol{S}_t = \boldsymbol{s}, \boldsymbol{A}_t = \boldsymbol{a}), \tag{2.9}$$

where $\mathbb{P}$ denotes the probability operator. The Markovian property of the transition model is that, for any time step $t$,

$$\mathbb{P}(\boldsymbol{S}_{t+1}|\boldsymbol{A}_t, \boldsymbol{S}_t, \ldots, \boldsymbol{A}_0, \boldsymbol{S}_0) = \mathbb{P}(\boldsymbol{S}_{t+1}|\boldsymbol{A}_t, \boldsymbol{S}_t). \tag{2.10}$$

Essentially, the Markovian property requires that a future state is independent of all past states given the current state. Assume $\mathcal{P}$ is time-homogeneous, i.e., for any two time steps $t_1$ and $t_2$,

$$\mathcal{P}_{t_1}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) = \mathcal{P}_{t_2}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}). \tag{2.11}$$

Then, we can drop the superscript $t$ and write the transition model as $\mathcal{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$. Note that when $\mathcal{S}$ is continuous, the transition model can be represented by a conditional probability density function.

Let $\pi : \mathcal{S} \to \mathcal{A}$ denote a deterministic policy for the MDP defined above. The action-value function for the MDP under policy $\pi$ is defined as follows:

$$Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | \boldsymbol{S}_0 = \boldsymbol{s}, \boldsymbol{A}_0 = \boldsymbol{a}; \pi\right], \tag{2.12}$$

where $\mathbb{E}$ denotes the expectation. The action-value function $Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$ is the expected cumulative discounted reward when the system starts from state $\boldsymbol{s}$, takes action $\boldsymbol{a}$, and follows policy $\pi$ thereafter. The maximum action-value function over all policies is defined as $Q(\boldsymbol{s},\boldsymbol{a}) = \max_{\pi} Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$. A policy $\pi$ is said to be optimal if $Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) = Q(\boldsymbol{s},\boldsymbol{a})$ for any $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$. In particular, the greedy policy with respect to $Q(\boldsymbol{s},\boldsymbol{a})$, defined as $\pi^*(\boldsymbol{s}) = \arg\max_{\boldsymbol{a}} Q(\boldsymbol{s},\boldsymbol{a})$, is an optimal policy (Sutton and Barto, 2018). The MDP is solved if we find $\pi^*$.

**Theorem 1.** *(Bertsekas and Tsitsiklis, 1996) The optimal action-value function $Q(\boldsymbol{s},\boldsymbol{a})$ satisfies the Bellman optimality equation:*

$$Q(\boldsymbol{s},\boldsymbol{a}) = \mathbb{E}[R_0] + \gamma \sum_{\boldsymbol{s}' \in \mathcal{S}} \mathcal{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \max_{\boldsymbol{a}' \in \mathcal{A}} Q(\boldsymbol{s}',\boldsymbol{a}'). \tag{2.13}$$

*Furthermore, there is only one $Q$ function that solves the Bellman optimality equation.*

The Bellman optimality equation is of central importance to solving the MDP. When both $\mathcal{S}$ and $\mathcal{A}$ are finite and $\mathcal{P}$ is known, model-based based algorithms such as the value iteration algorithm can be applied to solve the MDP (Sutton and Barto, 2018).

### Finding Optimal Learning Policy as MDP

We next formulate the problem of finding the optimal learning policy as an MDP. The state space $\mathcal{S}$ is the set of all attribute profiles which are constructed under the proposed hierarchical learning model. The action space $\mathcal{A}$ is defined to be the set of all learning materials $\mathcal{A} = \mathcal{L} = \{1, 2, \cdots, L\}$.

As discussed earlier, the transition model $\mathcal{P}$ satisfies the Markov property. Let $\boldsymbol{S}_t$ and $\boldsymbol{L}_t$ denote the state and action at time step $t$, respectively. Then, the transition model $\mathcal{P}_t(\boldsymbol{\alpha}'|\boldsymbol{\alpha}, l) = \mathbb{P}(\boldsymbol{S}_{t+1} = \boldsymbol{\alpha}'|\boldsymbol{S}_t = \boldsymbol{\alpha}, \boldsymbol{L}_t = l)$ is the probability of transitioning from state $\boldsymbol{\alpha}$ to state $\boldsymbol{\alpha}'$ after taking action $l \in \mathcal{L}$ at time step $t$. Denote $r = \mathcal{R}(\boldsymbol{\alpha}, l, \boldsymbol{\alpha}')$ which describes the reward obtained when the state transitions from $\boldsymbol{\alpha}$ to $\boldsymbol{\alpha}'$ after taking action $l$. The goal of reinforcement learning is to maximize the expected cumulative reward where a reward is a scalar feedback signal indicating how good the taken action is (Sutton and Barto, 2018). In this chapter, an optimal learning policy selects a learning material among available materials for learners so that each individual's total learning steps taken to master all attributes is minimized.

Q-learning Algorithm for Hierarchical Learning Model

The overall framework is illustrated in Figure 2.3, where the agent is the adaptive learning system that determines an action (i.e., learning material), sent to the environment (i.e., learners), which will then send the state (i.e., attribute profiles), and a reward signal back to the agent. Since both the state space and the action space are discrete, a classical data-driven RL algorithm—the Q-learning algorithm—can be applied to learn the optimal learning policy (Watkins and Dayan, 1992). RL is widely used in solving problems by interacting with the environment, without requiring an explicitly expressed MDP model (Sutton and Barto, 2018). The RL method has several advantages of finding the optimal learning policy in the adaptive learning system. First, since it does not require an explicit model to estimate the utility of taking actions in the environment (Kaelbling et al., 1996), the RL method can be an ideal solution for finding the best solution for an adaptive learning system, where how a learner' attribute profile changes after feeding a learning material is unknown. Second, the learning path with attribute hierarchy modeled by an HMM can be well-solved by the RL method. Third, the RL method searches for the long-term optimal solution which takes future rewards into consideration instead of simply choosing the best option at immediate step (Littman, 1994).

The Q-learning algorithm estimates an action value function—the so called Q-function—that is the expected cumulative discounted reward of a state-action pair. The action-value function under policy $\pi$ such that

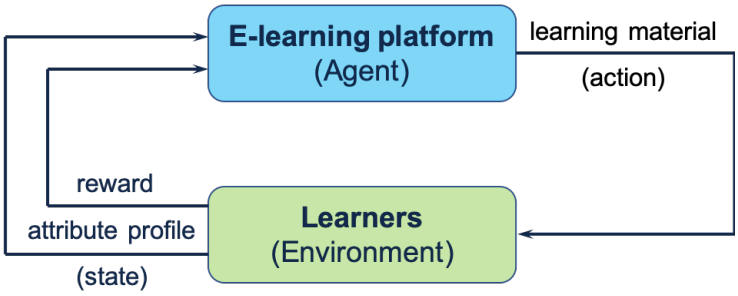


Figure 2.3: RL system in the optimal learning policy problem.

$l = \pi(\boldsymbol{\alpha})$, $\boldsymbol{\alpha} \in \mathcal{S}$ and $l \in \mathcal{A}$, is defined as follows:

$$Q^{\pi}(\boldsymbol{\alpha}, l) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t | \boldsymbol{S}_0 = \boldsymbol{\alpha}, \boldsymbol{L}_0 = l; \pi], \tag{2.14}$$

where $\mathbb{E}$ is the expectation operator, $\gamma \in [0, 1)$ is a discount-rate parameter, and $R_t$ is the reward obtained after taking action $\boldsymbol{A}_t$ at state $\boldsymbol{S}_t$, which is a random variable whose value is denoted as $r$. Note that discounting is an additional concept in RL algorithms. The discount rate $\gamma$ determines the present value of future rewards such that a reward received $t$ time steps in the future is worth only $\gamma^{t-1}$ times what it would be worth if it were received immediately (see, e.g., Sutton and Barto, 2018, for more details). Therefore, the agent (adaptive learning system) selects actions to maximize the sum of the discounted rewards it receives over the future.

Denote the maximum action-value function over all policies as $Q^*(\boldsymbol{\alpha}, l) = \max_{\pi} Q^{\pi}(\boldsymbol{\alpha}, l)$. Then the greedy policy regarding $Q^*(\boldsymbol{\alpha}, l)$ is an optimal policy denoted as $\pi^*(\boldsymbol{\alpha})$ such that $\pi^*(\boldsymbol{\alpha}) = \arg\max_l Q^*(\boldsymbol{\alpha}, l)$, $\forall \boldsymbol{\alpha} \in \mathcal{S}$ and $\forall l \in \mathcal{L}$. The MDP is solved if the optimal policy $\pi^*(\boldsymbol{\alpha})$ is found, or equivalently, the maximum action-value function $Q^*(\boldsymbol{\alpha}, l)$ is found. Note that $Q(\boldsymbol{\alpha}, l)$ satisfies the Bellman optimality equation (see, e.g., Sutton and Barto, 2018), defined as

$$Q^*(\boldsymbol{\alpha}, l) = \mathbb{E}[R_0] + \gamma \sum_{\boldsymbol{\alpha}' \in \mathcal{S}} \mathcal{P}(\boldsymbol{\alpha}' | \boldsymbol{\alpha}, l) \max_{l' \in \mathcal{L}} Q(\boldsymbol{\alpha}', l'), \tag{2.15}$$

which is the key in solving the MDP. Since both the state and action sets are discrete, the action-value function can be represented in a tabular form covering all possible state-action pairs $(\boldsymbol{\alpha}, l) \in \mathcal{S} \times \mathcal{L}$.

**Algorithm 1:** Q-learning Algorithm for Hierarchical Learning Model

**Data:** attribute profile (state) set $\{\boldsymbol{\alpha}\}$, action set $\mathcal{L}$, learning rate $\beta$, discount factor $\gamma$, decay rate for learning $\lambda_\beta$, decay rate for exploration $\lambda_\epsilon$, initial exploration probability $\epsilon$, weights for the reward function $w_1$, $w_2$, $d_1$, $d_2$

**Result:** Q function

Randomly initialize the value of $Q(\boldsymbol{\alpha}_0, l_0)$

Receive initial state $\boldsymbol{\alpha}_0$

**for** $t = 0, 1, \cdots$ **do**

Select $l_t \leftarrow \text{argmax}_l Q(\boldsymbol{\alpha}_t, l)$ with probability of $1 - \epsilon$ and otherwise randomly select $l_t$ with probability of $\epsilon$

Receive a new state $\boldsymbol{\alpha}_{t+1}$

Calculate $n_{\boldsymbol{\alpha}_t}$ as the number of mastered attributes at time step $t$ for the learner

Compute reward $r_t$ according to

$$r_t = \begin{cases} w_1 \times (n_{\boldsymbol{\alpha}_{t+1}} - n_{\boldsymbol{\alpha}_t}) - d_1 \times t, & \text{if } n_{\boldsymbol{\alpha}_{t+1}} > n_{\boldsymbol{\alpha}_t} \\ -w_2 \times (1 + n_{\boldsymbol{\alpha}_t} - n_{\boldsymbol{\alpha}_{t+1}}) - d_2 \times t, & \text{if } n_{\boldsymbol{\alpha}_{t+1}} \leq n_{\boldsymbol{\alpha}_t} \end{cases}$$

Calculate $Q$ value

$$Q(\boldsymbol{\alpha}_{t+1}, l_t) := Q(\boldsymbol{\alpha}_t, l_t) + \beta(r_t + \gamma \max_{l'} Q(\boldsymbol{\alpha}_{t+1}, l') - Q(\boldsymbol{\alpha}_t, l_t))$$

Update learning rate $\beta \leftarrow \beta \times \lambda_\beta$ and exploration rate $\epsilon \leftarrow \epsilon \times \lambda_\epsilon$

**end**

The Q-learning algorithm approximates $Q^*(\boldsymbol{\alpha}, l)$ by iteratively updating the action-value function $Q(\boldsymbol{\alpha}_{t+1}, l_t) := Q(\boldsymbol{\alpha}_t, l_t) + \beta(r_t + \gamma \max_{l'} Q(\boldsymbol{\alpha}_{t+1}, l') - Q(\boldsymbol{\alpha}_t, l_t))$, where $\beta$ is the learning rate which decays over iterations with a decay rate of $\lambda_\beta$ (see, e.g., Watkins, 1989, for more details). The detailed algorithm for the optimal learning policy problem is presented in Algorithm 1. The $\epsilon$-greedy exploration policy is adopted with a probability of $\epsilon$ to explore at the beginning and decayed later for exploitation, with the decay rate for exploration denoted as $\lambda_\epsilon$. In addition, the reward in Algorithm 1 is

$$r_t = \begin{cases} w_1 \times (n_{\boldsymbol{\alpha}_{t+1}} - n_{\boldsymbol{\alpha}_t}) - d_1 \times t, & \text{if } n_{\boldsymbol{\alpha}_{t+1}} > n_{\boldsymbol{\alpha}_t} \\ -w_2 \times (1 + n_{\boldsymbol{\alpha}_t} - n_{\boldsymbol{\alpha}_{t+1}}) - d_2 \times t, & \text{if } n_{\boldsymbol{\alpha}_{t+1}} \leq n_{\boldsymbol{\alpha}_t} \end{cases}, \tag{2.16}$$

where $n_{\boldsymbol{\alpha}_t}$ is the number of mastered attributes in the attribute profile $\boldsymbol{\alpha}_t$ at time step $t$, and $w_1$, $w_2$, $d_1$, $d_2$ are positive real numbers that reflect

the relative importance of two objectives–increasing the number of mastered attributes and decreasing the length of the learning episode. The reward decreases while the length of the learning episode (the entire learning process of a learner from beginning to mastery) increases, and the reward increases while the number of mastered attributes increases since our objective is to minimize the episode length for each learner to master all attributes. Note that $\boldsymbol{\alpha}_t$ is estimated by CDMs, and $n_{\boldsymbol{\alpha}_{t+1}}$ can be smaller than $n_{\boldsymbol{\alpha}_t}$ with the presence of estimation errors. The Q-learning algorithm proves to converge, that is, the $Q$ values converge to $Q^*$ with probability 1 if the learning rate is properly chosen and the state-action space is sufficiently explored (Watkins and Dayan, 1992).

## 2.4 SIMULATION STUDIES AND RESULTS

In this section, we apply the Q-learning algorithm to find the optimal learning policy with attribute profiles modeled by the hierarchical learning model in both cases, with or without the presence of measurement errors. We also investigate the impacts of various initial states.

### 2.4.1 Overview

The purpose of the simulation studies is to explore how the RL method performs in finding the optimal learning policy with the transition process of hierarchical attributes built in the framework of hierarchical learning model. The optimal learning policy found by the RL method is compared with a heuristic method. Since measurement errors always exist in the estimated attribute profiles using CDMs, it is important to evaluate the impact of estimation errors on the optimal learning policy found by the RL method. Different magnitudes of measurement errors are added to the true attribute profile representing the estimated ones. In addition, because of the fact that learners should be given different optimal learning paths according to their initial attribute profiles, whether the system can find the optimal learning policy for learners with different initial attribute profiles is examined.

As explained in "Hierarchical Learning Model" section, an attribute hierarchical structure can be represented by its paired attributes, the simulation study considers two attributes with linear hierarchical structure. Denote the two attributes as $A_1$ and $A_2$. Assume each attribute has three mastery levels

Table 2.3: State Space for Two Attributes with Three Levels.

| State | $A_1^{(1)}$ | $A_1^{(2)}$ | $A_1^{(3)}$ | $A_2^{(1)}$ | $A_2^{(2)}$ | $A_2^{(3)}$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 | 0 |
| 6 | 1 | 1 | 0 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 0 | 0 |
| 8 | 1 | 1 | 0 | 1 | 1 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 |

denoted as $A_1^{(1)}$, $A_1^{(2)}$, $A_1^{(3)}$, $A_2^{(1)}$, $A_2^{(2)}$, and $A_2^{(3)}$, respectively. $A_1^{(0)}$ or $A_2^{(0)}$ is used when the corresponding attribute is not mastered.

Assume $A_1$ is a prerequisite attribute of $A_2$, satisfying all assumptions in the section "Hierarchical Learning Model". An intuitive way to understand the hierarchical structure here is to assume $A_1$ to be the addition and $A_2$ to be the multiplication. The three mastery levels can be translated to beginner, intermediate and advanced level, while $A_1^{(0)}$ and $A_2^{(0)}$ indicate a learner has no knowledge of $A_1$ and $A_2$, respectively.

Assume six learning materials are available, three of which are beginner, intermediate and advanced level materials for attribute $A_1$, denoted as $l_1^{(1)}$, $l_1^{(2)}$, $l_1^{(3)}$, and the other three are for attribute $A_2$, denoted as $l_2^{(1)}$, $l_2^{(2)}$, $l_2^{(3)}$, intended for three mastery levels of each attribute. We thus construct the Markov decision process shown as a directed graph in Figure 2.4. Each circle represents a state. A full arrow shows a transition of attribute $A_1$ while a dotted arrow shows a transition of attribute $A_2$. Only one attribute can be improved in each learning step. If the learner acquires the attribute profile of $A_1^{(3)}A_2^{(3)}$, no more learning material will be provided and the learning process ends. The process satisfies the three assumptions in the hierarchy learning model. Note that the transition from a state to itself is neglected in the directed graph and can be easily calculated by Markov properties. The transition matrix, which is unknown to the environment and only applied to predict learners' next state, is constructed accordingly. The corresponding state space is shown in Table 2.3.

Figure 2.4 reveals the difference between the policy that only considers

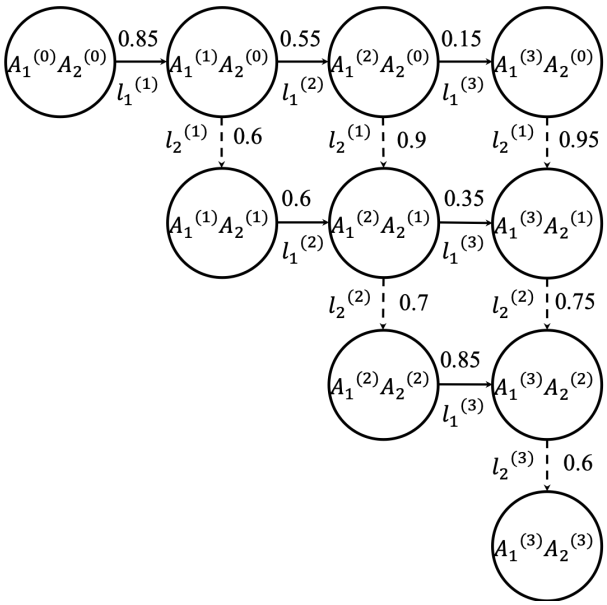


Figure 2.4: The directed graph of the Markov process for the attribute profile consisting of attribute $A_1$ and $A_2$.

immediate reward and the policy given by the RL method that takes future rewards into consideration. The learning materials related to the corresponding state transition are also presented in the figure. For instance, suppose a learner reaches the beginner level of the first attribute $A_1$ and has no knowledge of the second attribute $A_2$, i.e., in state $A_1^{(1)}A_2^{(0)}$. The beginner level material for attribute $A_2$, which is denoted as $l_2^{(1)}$, gives the shorter expected learning time at this step defined as $\sum_{t=1}^{\infty} t\mathbb{P}(\alpha_{s,t} = 1|\alpha_{s,0} = 0)$ which is $1/0.6 \approx 1.67$ [1]. However, although the intermediate level material for attribute $A_1$, denoted as $l_1^{(2)}$, brings relatively longer learning time, leading to a smaller reward at current step, the overall expected learning time of path through $A_1^{(2)}A_2^{(0)}$ to $A_1^{(2)}A_2^{(1)}$, which is $1/0.55 + 1/0.9 \approx 2.93$, is less than that through $A_1^{(1)}A_2^{(1)}$ to $A_1^{(2)}A_2^{(1)}$, which is $1/0.6 + 1/0.6 \approx 3.33$. As a result, although to learn beginner level attribute $A_2$ first is quicker at current step, it is not the long-term optimal learning path.

Since the estimated attribute profile is adopted instead of the true state in practice, an estimation error of 0.05 was added to the state, indicating there is a 5% probability that the estimated attribute profile is incorrect. In CDM research, the average pattern correct classification rate (PCCR) is usually larger than 95%. Therefore, an estimation error of 0.05 is large

[1]Expected value for a geometric distribution.

enough to show the reliability of the optimal learning policy. In addition, simulation results for cases with an estimation error ranging from 1% to 10% are included to show that the Q-learning algorithm is reliable and stable to find the optimal learning policy even with the presence of an estimation error. In practice, the attribute profiles are estimated and updated based on responses of test items.

The rest of parameters in the algorithm are set as follows. The initial learning rate is $\beta = 0.01$ and the discount factor is $\gamma = 0.99$, both of which are values widely used in machine learning algorithms. The initial value of the exploration probability is $\epsilon = 1$, i.e., the algorithm will choose an action randomly in the beginning. To stabilize the learning algorithm, the learning rate $\beta$ and the exploration probability $\epsilon$ need to be decreased gradually. Therefore, a decay rate $\lambda_\beta$ of 0.999 is applied for the learning rate $\beta$ and a decay rate $\lambda_\epsilon$ of 0.99 is used for the exploration probability $\epsilon$. After 5000 episodes (representing 5000 learners complete their learning processes), the learning rate $\beta$ decays to a value of 0.7% and the exploration probability $\epsilon$ decays to $1.50 \times 10^{-22}$. Weights for the reward are $w_1 = 2$, $w_2 = 1$, $d_1 = d_2 = 0.1$. Note that these weights indicate that the importance of increasing the number of attributes is more preferred than decreasing the length of the learning episode.

The Q-learning algorithm is trained in 5000 episodes to acquire a stable policy. After that, the trained model is applied in another 1000 episodes (e.g., new learners) and compared with a heuristic learning policy, which selects the next learning material that can improve the learner's mastery levels of both attributes in accordance with hierarchical learning model assumptions. For instance, if the learner's attribute profile is estimated to be $A_1^{(1)}A_2^{(0)}$, the learning material will be selected from beginner level material for attribute $A_2$ and intermediate level material for attribute $A_1$. The two methods are compared both with and without an estimation error.

Recall that the objective of the optimal learning policy is to minimize the total learning steps and the reward in the Q-learning algorithm is designed to increase when the total learning steps decrease. The evaluation criterion to compare the two methods is the episode reward such that the higher the episode reward is, the better learning policy the method finds. The standard deviation (SD) of rewards is also calculated and compared between the two methods as the smaller the SD is, the more stable the optimal learning policy

is. In addition, the number of episodes that the RL method takes to find the optimal learning policy is also examined. A short episode length indicates the method can find the optimal learning policy quickly.

Two simulation studies are conducted in this work. In the first simulation study, the initial states for all learners are $A_1^{(0)}A_2^{(0)}$, which means none of the learners have any knowledge of the two attributes. In the second simulation study, learners start with any mastery levels excluding $A_1^{(3)}A_2^{(3)}$. The second simulation study shows that as long as a learner has not fully mastered attributes specified in the adaptive learning system, no matter which level the learner begins with, the system can find the optimal learning policy for the learner.

### 2.4.2 Results

#### Learning Policy Comparison

Figures 2.5 and 2.6 present the rewards of the first 500 episodes when the RL method is trained, including both the immediate reward and the smoothed reward with a smoothing window of 20. Figure 2.5 shows that the reward becomes stable after 200 episodes under the RL method without estimation error, which means the method finds the optimal learning policy after self-training on 200 learners. The result indicates that the RL method finds the optimal learning policy quickly. After a 5% estimation error is added to the system, the Figure 2.6 presents that the RL method still finds the optimal learning policy after around 250 episodes.

Figures 2.7 and 2.8 give a comparison between the trained RL method and heuristic method applied in 1000 new episodes. No estimation error is added in Figure 2.7 while a 5% estimation error is added to both methods in Figure 2.8. Both figures show that the reward under the RL method is higher than the heuristic method. The smoothed reward of the RL method is significantly higher than that of the heuristic method in both with or without an estimation error.

Table 2.4 shows the overall mean and standard deviation of rewards and episode lengths in two methods. The RL method has much higher mean and lower standard deviation of rewards than the heuristic method, together with shorter episode lengths and smaller episode length standard deviation

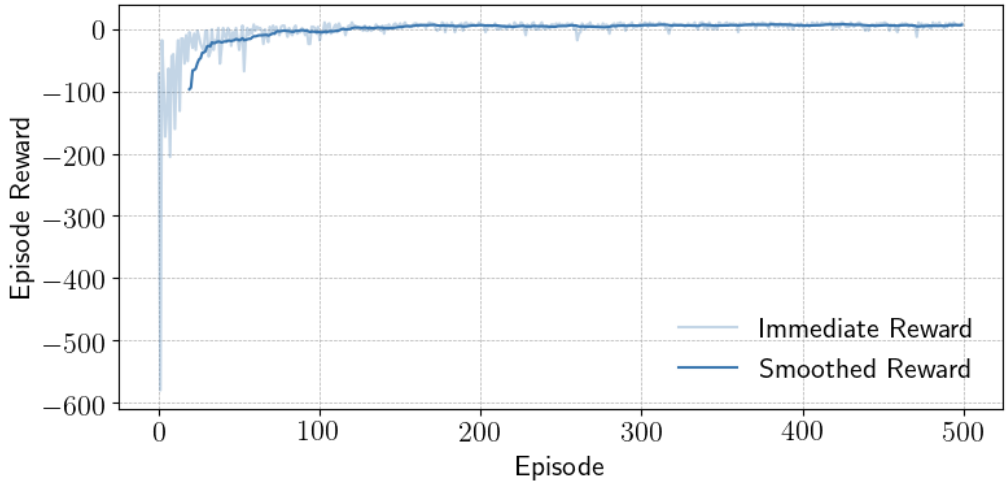


Figure 2.5: Rewards under the optimal learning policy without an estimation error.

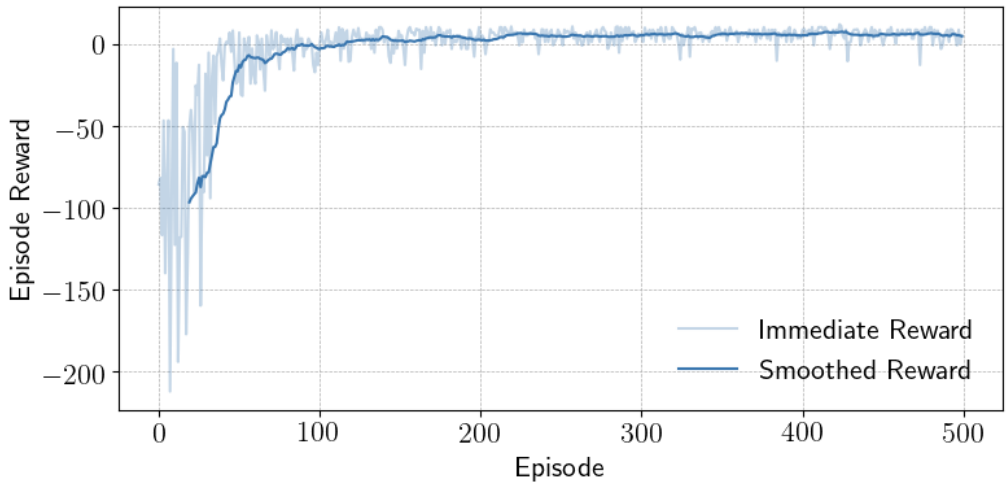


Figure 2.6: Rewards under the optimal learning policy with a 5% estimation error.

as well. It is worth noting that although the average episode length with a 5% estimation error is slightly higher than that without an estimation error, the difference is trivial.

Figure 2.9 gives a comparison between the RL method and heuristic method across 1000 episodes under 10 different estimation errors and no estimation error using the box plot. The figure shows that the average reward under the RL method is much higher than that under the heuristic method across the 11 conditions. In addition, the RL method also produces smaller

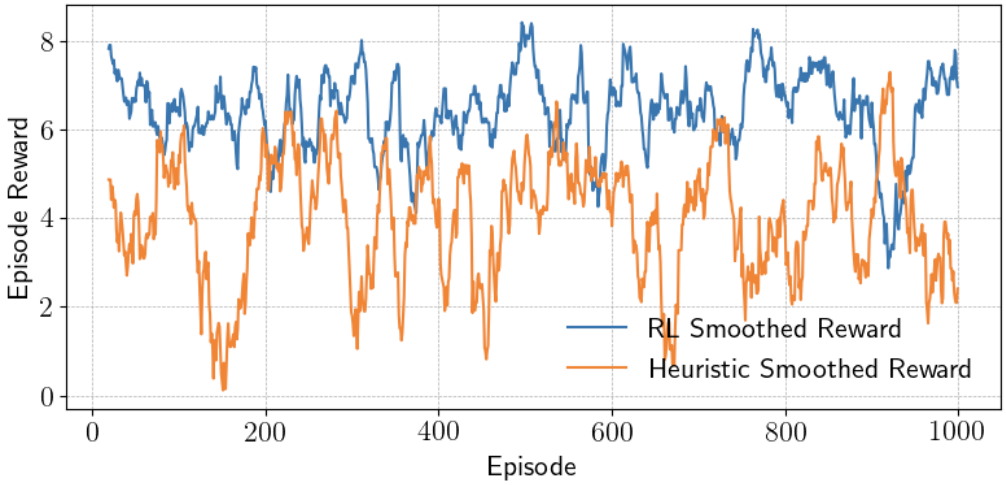


Figure 2.7: Smoothed rewards under the optimal learning policy learned via RL and the heuristic learning policy without an estimation error.


Figure 2.8: Smoothed rewards under the optimal learning policy learned via RL and the heuristic learning policy with a 5% estimation error.

standard deviation of rewards than the heuristic method. Although the standard deviation of the RL method tends to increase when the estimation error increases, it is still smaller than that of the heuristic method.

The simulation results shown above indicate that the RL method finds a better learning policy than the heuristic method. More importantly, the estimation error has negligible impact on the performance of the RL method in searching for the optimal learning policy.

Table 2.4: Mean and Standard Deviation (SD) of Rewards and Episode Lengths (EL).

| | Methods | RL | Heuristic |
|---|---|---|---|
| No Estimation Error | Reward mean | 6.43 | 3.99 |
| | Reward SD | 3.61 | 5.34 |
| | EL mean | 7.34 | 8.57 |
| | EL SD | 1.90 | 2.62 |
| 5% Estimation Error | Reward mean | 6.41 | 3.98 |
| | Reward SD | 3.60 | 5.37 |
| | EL mean | 7.73 | 9.01 |
| | EL SD | 2.07 | 2.74 |

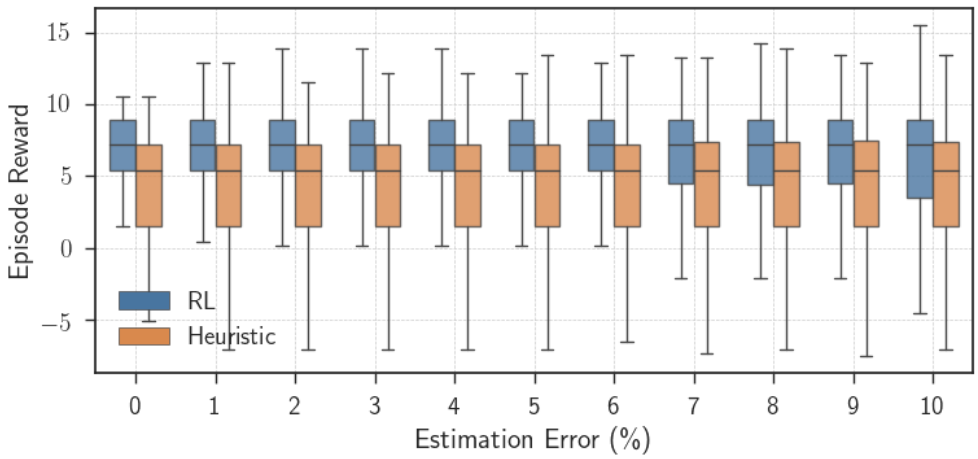


Figure 2.9: Comparison of rewards under the optimal learning policy learned via RL and the heuristic learning policy with estimation errors.

Impacts of Various Initial States

Figure 2.10 presents the smoothed rewards of the nine different initial states not including $A_1^{(0)}A_2^{(0)}$ with a smoothing window of 20. A 5% estimation error

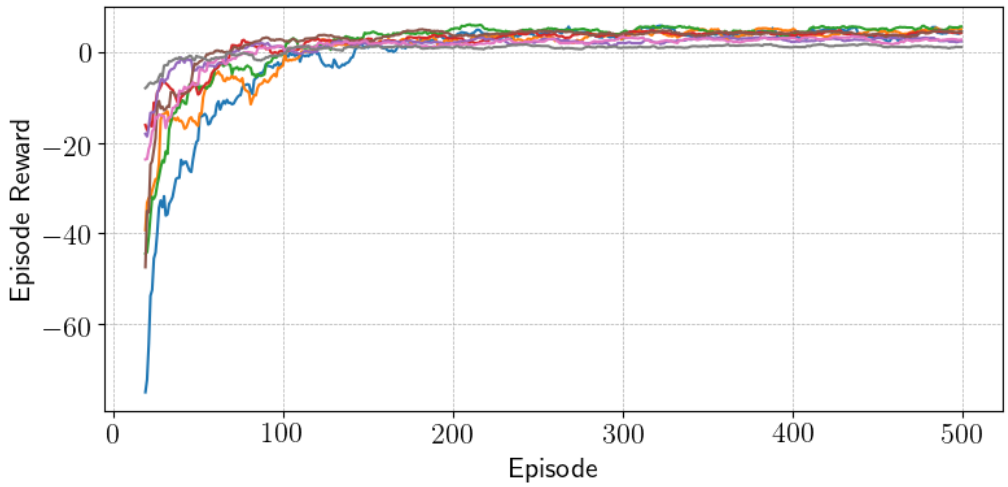


Figure 2.10: Smoothed rewards of different initial states under the optimal learning policy with a 5% estimation error.

is added to the system to simulate realistic cases. The result demonstrates that the RL method can quickly find the optimal learning policy for all learners regardless of the initial attributes. The algorithm converges after 200 episodes indicating that the optimal policy can be found after it is trained on only 200 learners. Therefore, once a learner's initial attribute is estimated by a set of items, the learner can follow the optimal learning policy to acquire new attributes with the fastest route provided by the system.

## 2.5 DISCUSSION

We proposed a hierarchical learning model that incorporates attribute hierarchy and mastery levels of attributes together in an adaptive learning system. The model follows the same form of discrete attributes and Q-matrix required by CDMs so that parameters and hidden states can be easily recovered and estimated. In addition, the transition process for the learning path given a learning material is formulated as an MDP. Then, a data-driven RL method is applied to finding the optimal learning policy on top of the hierarchical framework.

Simulation results suggest that the optimal design with the RL method outperforms the heuristic learning policy substantially with and without an estimation error. The mean and the standard deviation of the learning episode length achieved by the RL method is significantly smaller compared

to those obtained by the heuristic method. In addition, the RL method can find the optimal learning policy quickly for all learners with different initial attributes. As a result, learners with various levels of attributes will be assigned respectively the most appropriate materials at each step. In practice, a set of items needs to be given to learners after they finish each learning material and then, their attributes are estimated and updated based on their responses to the given items.

# CHAPTER 3

# ADAPTIVE LEARNING SYSTEMS WITH DEEP REINFORCEMENT LEARNING

## 3.1 INTRODUCTION

Designing the optimal learning path which selects the most appropriate learning materials to each individual learner based on their historical information has emerged as a promising and important topic in recent years, along with the widespread shift from traditional classroom teaching to adaptive learning systems (Means et al., 2009). Numerous challenges exist in developing an adaptive learning system, and the biggest challenges are: (i) how to optimally select materials for individual learners based on their current learning status; (ii) after being fed with a learning material, how much a learner's latent traits will improve is unknown and as a result the change cannot be characterized by a deterministic model; (iii) the learning policy should be dynamically trained as new learning materials are added; (iv) when the learners' data is not enough to train the most optimal learning policy, a relatively optimal policy is needed; (v) how to characterize learners' latent traits in continuous scales which provide more information than discrete levels of competences.

In previous studies, the proficiencies or latent traits were typically characterized as vectors of binary latent variables (Chen et al., 2018c; Li et al., 2018; Tang et al., 2019). However, it is important to consider the granularity of the latent traits in a complicated learning and assessment environment in which a knowledge domain consists of several fine-grained abilities. In some cases, it would be too simple to model learners' abilities as mastery or non-mastery. For example, when an item is designed to measure several latent traits and a learner regarded as mastering all related traits of the item cannot be assured to answer the item correctly. A possible reason is that the so-called mastery is not full mastery of a latent trait. By measuring learners' traits as continuous scales, the adaptive learning system can be designed to help learners to learn and improve until they reach the target levels of certain abilities so that the learners can achieve target scores in assessments. Especially in practice, most assessments are designed to measure learners' latent traits (McGlohen and Chang, 2008). In such scenarios, it is better

to use a continuous scale to measure the latent traits as the item response theory (IRT) does. In this chapter, we will develop an adaptive learning system that estimate learners' abilities using measurement models in order to provide them with most appropriate materials for further improvements.

Existing research studies have focused on modeling learners' learning paths (Chen et al., 2018b; Wang et al., 2018), accelerating learners' memory speed (Reddy et al., 2017), providing model-based sequence recommendation (Chen et al., 2018c; Lan and Baraniuk, 2016; Xu et al., 2016), tracing learners' concept knowledge state transitions over time (Lan et al., 2014), and selecting materials for learners optimally based on model-free algorithms (Li et al., 2018; Tang et al., 2019). However, explicit models are typically needed to characterize learners' learning progresses in these studies. While there exist research studies that aim to find the optimal learning policy which chooses the most appropriate learning materials for learners using model-free algorithms, they all assume discrete latent traits. In addition, when the number of learners is too small for the system to learn an optimal policy, these algorithms are not applicable. This work studies the important, yet less addressed adaptive learning problem—the problem of finding the optimal learning policy—based on continuous latent traits, and applies machine learning algorithms to deal with the tackle challenges such as only a small number of learners available in historical data.

In this chapter, we formulate the adaptive learning problem as a Markov decision process (MDP), in which the state is the profile of (continuous) latent traits of a learner, the action is the (discrete) learning material given to the learner. Yet, the state transition model is unknown in practice, thus making the MDP unsolvable using conventional model-based algorithms such as the value iteration algorithm (Sutton and Barto, 2018). To solve the issue, we apply a data-driven, model-free deep reinforcement learning (DRL) algorithm, the so-called deep Q-learning algorithm, to search for the optimal learning policy. The data-driven DRL algorithm is a class of machine learning algorithms that solve an MDP by learning an optimal policy represented by neural networks from a sequence of state transitions directly when the transition model itself is are unknown (François-Lavet et al., 2018). DRL algorithms have been widely applied in solving a variety of problems in many different fields such as playing Atari games (Mnih et al., 2015), bidding and pricing in electricity market (Xu et al., 2019), manipulating robotics (Gu

et al., 2017), and localizing objects (Caicedo and Lazebnik, 2015). We refer interested readers to François-Lavet et al. (2018) for a detailed review on the theories and applications of DRL. Therefore, the adaptive learning system is embedded with the well-developed measurement models and the data-driven DRL algorithm so as to be more flexible.

However, a deep Q-learning algorithm typically requires a large amount of state transition data so as to find an optimal policy, which may be difficult to obtain in practice. To cope with the challenge of insufficient state transition data, we develop a transition model estimator that emulates the learner's learning process using neural networks. The transition model that is fitted using available historical transition data can be used in the deep Q-learning algorithm to further improve its performance with no additional cost.

The purpose of this chapter is to develop a *fully adaptive* learning system in which (i) the learning material given to a learner is based on her/his continuous latent traits that indicate the levels of certain abilities, and (ii) the learning policy that maps the learner's latent traits to the learning materials is found adaptively with minimal assumption on the learners' learning process. First, an MDP formulation for the adaptive learning problem by representing latent traits in a continuum is developed. Second, a model-free DRL algorithm—the deep Q-learning algorithm—is applied, to the best of our knowledge, for the first time, in solving the adaptive learning problem. Third, a neural network based transition model estimator is developed, which can greatly improve the performance of the deep Q-learning algorithm when the number of learners is inadequate. Last, some interesting simulation studies are conducted to serve as demonstration cases for the development of adaptive learning systems.

## 3.2 PRELIMINARIES

In this section, we give a brief introduction on measurement models for continuous latent traits, which is an important component in adaptive learning systems. The representation of learners' latent traits and assumptions on them are also presented.

### 3.2.1 Measurement Models

In an adaptive learning system, a test is given to a learner/student after each learning cycle. The learner's responses to the test items are collected by the system and her/his latent traits are estimated using measurement models, specifically IRT models (Rasch, 1960; Lord et al., 1968).

An appropriate IRT model needs to be chosen based on the test's features such as the test's dimensional structure (Zhang, 2013) and its response categories. To be more specific, in the case when item responses are recorded as binary values indicating correct or incorrect answers, the test that evaluates only one latent trait will use the unidimensional item response theory IRT models (Rasch, 1960; Birnbaum, 1968; Lord, 1980), whereas tests that associate more than one trait will use the multidimensional item response theory (MIRT) models (Reckase, 1972; Mulaik, 1972; Sympson, 1978; Whitely, 1980). When item responses have more than two categories, polytomous IRT models such as the partial credit model (Masters, 1982), the generalized partial credit model (Muraki, 1992), and the graded response model (Samejima, 1969) are used for unidimensional case. Their extensions can be applied in multidimensional cases.

The basic representation of an IRT model is expressed as

$$\mathbb{P}(U = u|\boldsymbol{\theta}) = f(\boldsymbol{\theta}, \boldsymbol{\eta}, u), \tag{3.1}$$

where $\mathbb{P}$ denotes probability, $U$ is a random variable representing the score on the test item, $u$ is the possible value of $U$, $\boldsymbol{\theta}$ is a vector of parameters describing the learner's latent traits, $\boldsymbol{\eta}$ is a vector of parameters indicating the characteristic of the item, and $f$ denotes a function that maps $\boldsymbol{\theta}, \boldsymbol{\eta}, u$ to a probability in $[0, 1]$. As pointed out in Ackerman et al. (2003), many educational tests are inherently multidimensional. Therefore, we will use the MIRT as the intrinsic model to build up the adaptive learning system. As an illustration, the multidimensional two-parameter logistic IRT (M2PL) model is given by

$$\mathbb{P}(U_{ij} = 1|\boldsymbol{\theta}_i, \boldsymbol{a}_j, d_j) = \frac{e^{\boldsymbol{a}_j^\top \boldsymbol{\theta}_i + d_j}}{1 + e^{\boldsymbol{a}_j^\top \boldsymbol{\theta}_i + d_j}}, \tag{3.2}$$

where $U_{ij}$ is the response given by $i^{th}$ test taker to $j^{th}$ item, $\boldsymbol{\theta}_i = [\theta_{i1}, \theta_{i2}, \cdots, \theta_{iD}]^\top$

is a vector in $\mathbb{R}^D$ describing a set of $D$ latent traits, $\boldsymbol{a}_j$ is a vector of $D$ discrimination parameters for the $j^{th}$ item, indicating the relative importance of each trait in correctly answering the item, and the intercept parameter $d_j$ is a scalar for item $j$. An applicable item $j$ takes each element of $\boldsymbol{a}_j$ to be non-negative. Therefore, as each element's value of $\boldsymbol{\theta}_i$ increases, the probability of correct response increases.

With an online calibration design, an accurately calibrated item bank can be acquired using previous learners' response data for an adaptive learning system without large pretest subject pools (Makransky and Glas, 2014; Zhang and Chang, 2016). After item parameters are pre-calibrated, a variety of latent trait estimation methods can be applied to estimate learners' abilities. Conventional methods such as maximum likelihood estimation (Lord et al., 1968), weighted likelihood estimation and Bayesian methods (e.g. expected a posteriori estimation (EAP), maximum a posteriori (MAP)) can accurately estimate latent traits in MIRT models. Their variations are also extended for estimating the latent traits in multiple dimensions. Many latent trait estimation methods result in a bias on the order of as small as $O(n^{-1})$, where $n$ denotes test length, while approaches that further reduce the bias as well as the variance of estimates have also been identified and proposed (Firth, 1993; Tseng and Hsu, 2001; Wang, 2015; Warm, 1989; Zhang et al., 2011).

### 3.2.2 Assumptions

Denoted $\boldsymbol{\theta}^{(t)} = [\theta_1^{(t)}, \cdots, \theta_D^{(t)}]^\top$ as learner's latent traits at time step $t$, where $D$ is the number of dimensions. Throughout this chapter, we make the following simplifying yet practical assumptions:

**A**3.1. No retrogression exists in latent traits. That is, $\theta_d^{(t+1)} \geq \theta_d^{(t)}$, $\forall d \in \{1, \cdots, D\}$.

**A**3.2. The number of learning materials is finite.

## 3.3 ADAPTIVE LEARNING PROBLEM

In this section, we first describe the adaptive learning problem and then formulate this problem as an MDP.

### 3.3.1 Problem Statement

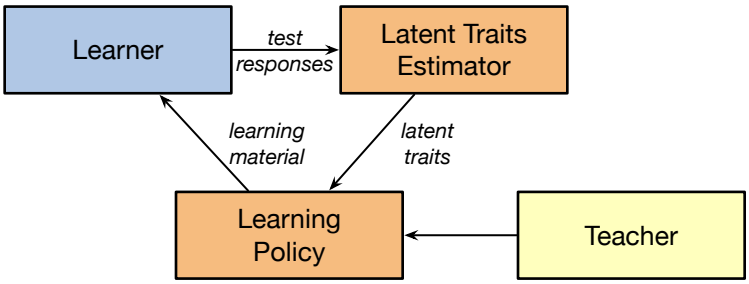


Figure 3.1: Conventional adaptive learning system.

A conventional adaptive learning system is illustrated in Figure 3.1. Such an adaptive learning system is typical in traditional classrooms and online courses like Massive Open Online Courses (MOOCs) (Lan and Baraniuk, 2016). In the adaptive learning system, the learner takes some learning materials to improve her/his latent traits. After the learner finishes learning the materials, a test or homework is assigned to the learner. Then, the learner's latent traits are estimated. Based on the estimated latent traits, the learning system adaptively determines the next learning material for the learner, which may be one of many forms including a textbook chapter, a lecture video, an interactive task, an instructor support, or an instruction pace. Such cyclic learning process continues until the learner's latent traits reach or are close to a prespecified levels of proficiency.

The tests in an adaptive learning system can be computerized adaptive testing (CAT). The CAT is a test mode that administers tests adapted to test takers' trait levels (Chang, 2015). It provides more accurate trait estimates with much smaller number of items (Weiss, 1982) by sequentially selecting and administering items tailored to each individual learner. Therefore, a relatively short test can assess learners' latent traits with high accuracy.

Conventionally, the learning policy (or plan) is provided by a teacher as illustrated in Figure 3.1. As aforementioned, however, it is too expensive for teachers to make an individualized adaptive learning policy for each learner. In this chapter, we use a DRL algorithm to search for an optimally individualized adaptive learning policy for each learner. The algorithm selects the most appropriate learning material among all available materials for each learner based on her/his provisional estimated latent traits that are obtained from

her/his learning history and performances in tests. The adaptive selection of learning materials guarantees the learner reaches a prespecified proficiency level in a shortest number of learning cycles or reaches proficiency level as high as possible in a fixed number of learning cycles. That is, instead of resorting to an experienced teacher for the construction of a learning policy as illustrated in Figure 3.1, we will develop a systematic method to enable the adaptive learning system to discover an optimal learning policy from the data that have been collected, which include historical learning materials, test responses, and estimated latent traits, etc.

### 3.3.2 Markov Decision Process Formulation

We next formulate the adaptive learning problem as an MDP. The same notation of MDP is adopted here as in Chapter 2.

*State Space:* Define the vector of parameters describing the learner's latent traits as the state, i.e., $\boldsymbol{s} = \boldsymbol{\theta}$, which has $D$ continuous variables, where $D$ represents the dimension of the latent traits. For the simplicity of the algorithm construction in the following, the state space is defined as $\mathcal{S} = [0,1]^D$ when each element of $\boldsymbol{\theta}$ satisfies $\theta \in [0,1]$, in which a smaller value of $\theta$ indicates a lower ability and a larger value indicates a higher ability. Although a latent trait variable is typically defined on $\mathbb{R}$ in IRT, a closed interval, say $[-5,5]$, is used as the range of a latent trait variable in practice. Let $h_d$ be the prespecified target proficiency level of the $d^{\text{th}}$ latent trait, which is the level the learners try to reach, where $d = 1,\ldots,D$. Because of the fact that there is a bijection between $[-5,h_d]$ and $[0,1]$, an estimated trait $\theta \in [-5,h_d]$ can be directly transformed into the scale of $[0,1]$. Thus, without loss of generality, we consider the state space as $\mathcal{S} = [0,1]^D$.

*Action Space:* Let the learning materials available in the adaptive learning system be indexed by $1,2,\cdots,L$. The action $\boldsymbol{a}$ in the adaptive learning system is represented by $l$, indicating the index of a learning material, which is discrete, and the action space is $\mathcal{A} = \{1,\cdots,L\}$.

*Reward Function:* Recall that the objective of the adaptive learning system is to minimize the learning steps it takes before a learner's latent traits reach the maximum, i.e., for $\boldsymbol{\theta}$ to reach $\mathbf{1}_D$, where $\mathbf{1}_D$ is an all-ones vector in $\mathbb{R}^D$.

As such, the reward function is defined as follows:

$$r = \mathcal{R}(\boldsymbol{s}, l, \boldsymbol{s}') = \begin{cases} -1, & \text{if } ||\boldsymbol{s}' - \mathbf{1}_D||_\infty < 10^{-3}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

where $||\cdot||_\infty$ indicates the infinite norm. Intuitively, the sum of rewards over one episode (the entire learning process of a learner) is to the negative of the total steps a learner takes before all of her/his latent traits are very close to 1, which indicates that the learner has reached target levels of all prespecified abilities.

*Transition Model:* The probability distributions of the latent trait as well as the change of trait are unknown. As a result, the transition model $\mathcal{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ is not known a priori.

Based on this MDP formulation, the adaptive learning problem is essentially to find an optimal learning policy, denoted by $\pi^* : \mathcal{S} \to \mathcal{A}$, that determines the action (learning material selection) based on the state (latent traits), such that the expected cumulative discounted reward is maximized. Note that the larger the expected cumulative discounted reward is, the less the total learning steps a learner takes to reach the target level(s) of an ability/abilities is. Since the transition model $\mathcal{P}$ is unknown, the MDP cannot be solved using model-based algorithms such as the value iteration algorithm. We will resort to a data-driven, model-free DRL algorithm to solve it in the next section.

## 3.4 OPTIMAL LEARNING POLICY DISCOVERY ALGORITHM

In this section, we solve the adaptive learning problem by using the deep Q-learning algorithm, which can learn the action-value function directly from historical transition data without knowing the underlying transition model. To utilize the available transition information more efficiently, we further develop a transition model estimator and use it to train the deep Q-learning algorithm.

### 3.4.1 Action-Value Function As Deep Q-Network

Recall that the optimal learning policy can be readily obtained if we know the action-value function. When the state is continuous and the action is

discrete, which is the case in the adaptive learning problem, the action-value function $Q(\boldsymbol{s}, l)$ cannot be exactly represented in a tabular form. In such cases, the action-value function can be approximated by some functions, such as linear functions (Sutton and Barto, 2018) or artificial neural networks (simply referred to as neural networks) (Mnih et al., 2015). In the former case, the approximate action-value function is represented as an inner product of the parameter vector and a feature vector that is constructed from the state. It is important to point out the choice of the features is critical to the performance of the approximate action-value function. Meanwhile, neural networks are capable of extracting useful features from the state directly, and have stronger representation power than linear functions (Goodfellow et al., 2016).

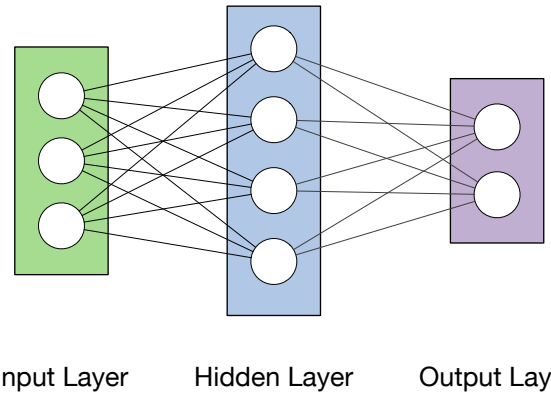


Figure 3.2: An illustrative neural network with one hidden layer.

As an example for neural networks, Figure 3.2 shows an illustrative neural network that consists of an input layer that has 3 units, a hidden layer that has 4 units, and an output layer with 2 units. Let $\boldsymbol{x} = [x_1, x_2, x_3]^\top$, $\boldsymbol{h} = [h_1, h_2, h_3, h_4]^\top$, and $\boldsymbol{y} = [y_1, y_2]^\top$ denote the vectors that come out of the input layer, the hidden layer, and the output layer, respectively. In the neural network, the output of one layer is the input for the next layer. To be more specific, $\boldsymbol{h}$ can be computed from $\boldsymbol{x}$, and $\boldsymbol{y}$ can be computed from $\boldsymbol{h}$ as follows:

$$\boldsymbol{h} = \phi(\boldsymbol{W}_{hx}\boldsymbol{x} + \boldsymbol{b}_h), \tag{3.4}$$

$$\boldsymbol{y} = \boldsymbol{W}_{yh}\boldsymbol{h} + \boldsymbol{b}_y, \tag{3.5}$$

where $\boldsymbol{W}_{hx} \in \mathbb{R}^{4\times 3}$ and $\boldsymbol{W}_{yh} \in \mathbb{R}^{2\times 4}$ are two weight matrices, $\boldsymbol{b}_h \in \mathbb{R}^4$

and $\boldsymbol{b}_y \in \mathbb{R}^2$ are two bias vectors, and $\phi(\cdot)$ is the so-called activation function, which is applied to its argument element-wise. A popular choice of the activation function $\phi$ is the rectifier, i.e., $\phi(x) = \max(x, 0)$. Conceptually, we can write the output $\boldsymbol{y}$ as a function of $\boldsymbol{y} = \varphi(\boldsymbol{x})$, where $\varphi(\cdot)$ is parameterized by $\boldsymbol{W}_{hx}$, $\boldsymbol{W}_{yh}$, $\boldsymbol{b}_h$, and $\boldsymbol{b}_y$, which can be collectively denoted as a parameter vector $\boldsymbol{w}$. Given a set of input-output values denoted by $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) : i = 1, \cdots, M\}$, the optimal value of $\boldsymbol{w}$ can be found by solving the following problem:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{M} ||\varphi(\boldsymbol{x}^{(i)}) - \boldsymbol{y}^{(i)}||^2, \tag{3.6}$$

where $||\cdot||$ is the $L_2$-norm. Problem (3.6) can be solved by using gradient descent algorithm or its variants, in which the gradient of the objective function with respect to $\boldsymbol{w}$ can be computed using the famous backpropagation technique. Neural networks can also be trained using a variety of other optimization algorithms such as Adam and RMSProp (see, Goodfellow et al., 2016). Note that there may be several hidden layers and the more hidden layers there are, the deeper the neural network is. We refer interested readers to Goodfellow et al. (2016) for a more comprehensive details about neural networks.

Recall that in the adaptive learning problem, the state is continuous in $[0, 1]^D$, while the action is discrete $\mathcal{A} = \{1, \cdots, L\}$. The approximate action-value function, denoted by $\hat{Q}(\boldsymbol{s}, l)$, can be represented using a neural network as follows. The input layer is the state $\boldsymbol{s}$, or equivalently, the latent trait vector $\boldsymbol{\theta}$, which has $D$ units. The output has $L$ units, each of which corresponds to the action-value for one action. To more be specific, the $\ell^{\text{th}}$ unit in the output layer gives $\hat{Q}(\boldsymbol{s}, l = \ell)$, i.e., the action-value for state $\boldsymbol{s}$ and action $\ell$. The number of hidden layers and the number of units in each hidden layer can be determined through simulation, which is to be detailed in the numerical simulation section. Such a neural network is also referred to as a deep Q-network (DQN) (Mnih et al., 2013). Let $\boldsymbol{w}$ denote the parameter vector of the DQN, which includes all weights and biases in the DQN. To emphasize that $\hat{Q}(\boldsymbol{s}, l)$ is parameterized by $\boldsymbol{w}$, we write $\hat{Q}(\boldsymbol{s}, l)$ as $\hat{Q}(\boldsymbol{s}, l; \boldsymbol{w})$.

Once we have $\hat{Q}(\boldsymbol{s}, l; \boldsymbol{w})$, the optimal learning policy becomes readily available, which is $\pi^*(\boldsymbol{s}) = \arg\max_l \hat{Q}(\boldsymbol{s}, l; \boldsymbol{w})$. Then, the "Teacher" block

in Figure 3.1 can be replaced with the DQN as shown in Figure 3.3.

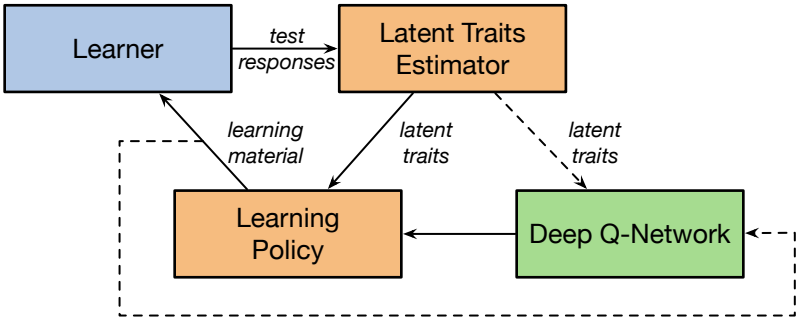


Figure 3.3: Adaptive adaptive learning system with DQN.

### 3.4.2 Learning Policy Discovery with Deep Q-Learning

The parameters of the DQN can be learned from the the sequence of latent traits and learning materials using the deep Q-learning algorithm proposed by Mnih et al. (2013). The optimal value of the parameter vector of the DQN, $\boldsymbol{w}$, can be found by minimizing the mean squared error between the approximate action-value function and the true action-value function:

$$\min_{\boldsymbol{w}} \mathbb{E}[(\hat{Q}(\boldsymbol{S}, \boldsymbol{A}; \boldsymbol{w}) - Q(\boldsymbol{S}, \boldsymbol{A}))^2]. \tag{3.7}$$

However, solving (3.7) is extremely difficult if not impossible since both $Q(\boldsymbol{S}, \boldsymbol{A})$ and the transition model are unknown and thus, the expectation of the mean squared error cannot be computed. The deep Q-learning algorithm adopts two measures to cope with these challenges. First, the expectation is replaced with the sample average that can be computed from a set of historical transitions, denote by $\mathcal{M} = \{(\boldsymbol{s}, l, r, \boldsymbol{s}') : \boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}, l \in \mathcal{A}\}$, with $|\mathcal{M}| = M$, where $|\cdot|$ denotes the cardinality of a set. That is, (3.7) is now replaced by the following problem:

$$\min_{\boldsymbol{w}} \sum_{(\boldsymbol{s},l,r,\boldsymbol{s}')\in\mathcal{M}} (\hat{Q}(\boldsymbol{s}, l; \boldsymbol{w}) - Q(\boldsymbol{s}, l))^2, \tag{3.8}$$

At time step $t$, the parameter vector is updated using the gradient descent algorithm as follows:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha \sum_{(\boldsymbol{s},l,r,\boldsymbol{s}')\in\mathcal{M}} (\hat{Q}(\boldsymbol{s},l;\boldsymbol{w}_t) - Q(\boldsymbol{s},l)) \frac{\partial \hat{Q}(\boldsymbol{s},l;\boldsymbol{w}_t)}{\partial \boldsymbol{w}}, \tag{3.9}$$

where $\alpha > 0$ is the learning rate and $\boldsymbol{w}^{(t)}$ denotes the value of $\boldsymbol{w}$ at time step $\boldsymbol{w}$. Second, the unknown $Q(\boldsymbol{s},l)$ is further substituted by $r+\gamma \max_{l'} \hat{Q}(\boldsymbol{s}',l';\boldsymbol{w}_t)$ based on the Bellman optimality equation in (2.15). Note that when $||\boldsymbol{s}' - \mathbf{1}_D||_\infty < 10^{-3}$, which indicates the learning process has ended, $Q(\boldsymbol{s}',l') = 0$. Therefore, (3.9) is now becomes

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha \sum_{(\boldsymbol{s},l,r,\boldsymbol{s}')\in\mathcal{M}} (\hat{Q}(\boldsymbol{s},l;\boldsymbol{w}_t) - y) \frac{\partial \hat{Q}(\boldsymbol{s},l;\boldsymbol{w}_t)}{\partial \boldsymbol{w}}, \tag{3.10}$$

where

$$y = \begin{cases} r, & \text{if } ||\boldsymbol{s}' - \mathbf{1}_D||_\infty < 10^{-3}, \\ r + \gamma \max_{l'} \hat{Q}(\boldsymbol{s}',l';\boldsymbol{w}_t), & \text{otherwise.} \end{cases} \tag{3.11}$$

The detailed deep Q-learning algorithm that is used to search the optimal parameter vector for the DQN is presented in Algorithm 2, where one episode represents a complete learning process of one learner and the number of episodes is the number of learners. In order to obtain a good approximate action-value function, the state-action space needs to be sufficiently explored. To achieve this, the so-called $\epsilon$-greedy exploration is adopted in the deep Q-learning algorithm. Specifically, at time step $t$, a random action $l_t$ is selected with probability $\epsilon_t$, and a greedy action $l_t = \max_l \hat{Q}(\boldsymbol{s}_t,l;\boldsymbol{w}_t)$ is with probability $1-\epsilon_t$. In this chapter, we adaptively decay $\epsilon_t$ from $\overline{\epsilon}$ to $\underline{\epsilon}$ in $\tau_\epsilon$ time steps. In addition, the parameter vector is updated at each time step using a set of transitions $\mathcal{M}$ that is resampled from the historical transitions denoted by $\mathcal{H}$ with $|\mathcal{H}| = H$ so as to reduce the bias that may be caused by the samples.

### 3.4.3 Transition Model Estimator

The deep Q-learning algorithm requires a sufficiently large historical transition data in order to find a good approximate of the action-value function,

**Algorithm 2:** Deep Q Learning Algorithm for Adaptive Learning Problem

**Data:** $\gamma, l, \overline{\epsilon}, \underline{\epsilon}, \tau_\epsilon, M, E$
**Result:** $\boldsymbol{w}$
Randomly initialize $\boldsymbol{w}$ and set total time step counter $\tau = 0$
**for** *episode* $= 1, \cdots, E$ **do**
  Receive initial state $\boldsymbol{s}_0$
  **for** $t = 0, 1, \cdots,$ **do**
    Compute $\epsilon_t = \overline{\epsilon} - (\overline{\epsilon} - \underline{\epsilon}) \times \min(\tau/\tau_\epsilon, 1)$ and increase $\tau$ by 1
    With probability $\epsilon_t$ select a random action $l_t$ otherwise select $l_t = \max_l \hat{Q}(\boldsymbol{s}_t, l; \boldsymbol{w}_t)$
    Send the learning material determined by $l_t$ to the student
    Given the student a test and collect test response
    Receive new state $\boldsymbol{s}_{t+1}$ estimated from test response by latent trait estimator
    Compute reward $r_t$ according to

$$r_t = \begin{cases} -1, & \text{if } ||\boldsymbol{s}_{t+1} - \mathbf{1}_D||_\infty \geq 10^{-3} \\ 0, & \text{otherwise} \end{cases}$$

    Store transition $(\boldsymbol{s}_t, l_t, r_t, \boldsymbol{s}_{t+1})$ into $\mathcal{H}$
    Sample $M$ transitions from $\mathcal{H}$ and store them into $\mathcal{M}$
    Update $\boldsymbol{w}$ according to

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha \sum_{(\boldsymbol{s}, l, r, \boldsymbol{s}') \in \mathcal{M}} (\hat{Q}(\boldsymbol{s}, l; \boldsymbol{w}_t) - y) \frac{\partial \hat{Q}(\boldsymbol{s}, l; \boldsymbol{w}_t)}{\partial \boldsymbol{w}}$$

    where

$$y = \begin{cases} r, & \text{if } ||\boldsymbol{s}' - \mathbf{1}_D||_\infty < 10^{-3} \\ r + \gamma \max_{l'} \hat{Q}(\boldsymbol{s}', l'; \boldsymbol{w}_t), & \text{otherwise} \end{cases}$$

    **if** $||\boldsymbol{s}' - \mathbf{1}_D||_\infty < 10^{-3}$, break
  **end**
**end**

based on which the learning policy is then derived. However, we may not be able to obtain adequate transitions due to several reasons including the lack of adequate learners, and the long time it takes to acquire an individual learner's learning path (transitions). Thus, it is more desirable to develop an adaptive learning system which can efficiently discover the optimal learning policy after training on a relatively small number of learners. To this end, we develop a transition model estimator which emulates the learning behavior

of learners. Specifically, the transition model estimator can take a state $\boldsymbol{s}$ and an action $l$ as inputs, and output the next state $\boldsymbol{s}'$. This can be cast as a supervised learning task, (a regression task), which can be solved using neural networks. The input layer of the neural network that represents the transition model is a pair of state and action, and the output layer is the next state. The number of hidden layers can be adjusted through the parameter tuning process (see, e.g., Goodfellow et al., 2016, for more details).

Conceptually, we can write the neural network that represents the transition model as $\boldsymbol{s}' = \psi(\boldsymbol{s}, l)$, the parameter vector of which is denoted by $\boldsymbol{v}$. The optimal value of $\boldsymbol{v}$ can be found by solving the following problem using the backpropagation algorithm:

$$\min_{\boldsymbol{v}} \sum_{(\boldsymbol{s},l,r,\boldsymbol{s}')\in\mathcal{H}} ||\psi(\boldsymbol{s}, l) - \boldsymbol{s}'||^2, \tag{3.12}$$

where $\mathcal{H}$ is the set of historical transition (data).

The adaptive learning system with the DQN and a transition model estimator is shown in Figure 3.4, where the DQN is trained against the transition model, instead of the actual learners.

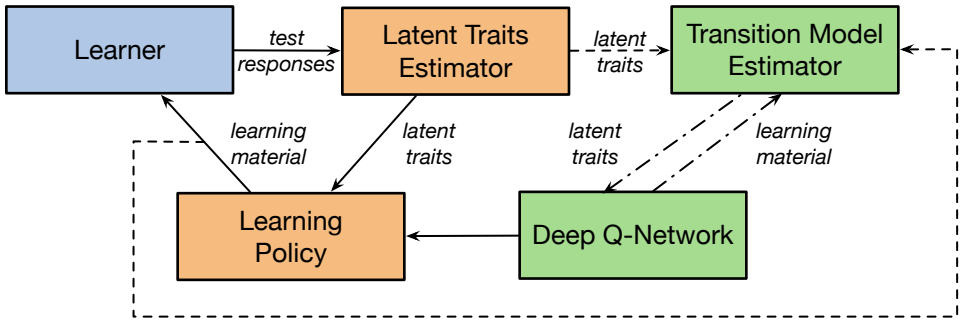


Figure 3.4: Adaptive adaptive learning system with DQN and transition model estimator.

## 3.5 SIMULATION STUDIES AND RESULTS

In this section, we show the performance of the adaptive learning system with and without the transition model estimator, and also investigate the impacts of latent trait estimation errors through two simulation studies.

### 3.5.1 Simulation Overview

Consider a group of learners in a two-dimensional assessment and a learning environment with three sets of learning materials. We model the group of learners as a homogeneous MDP. Recall that $\boldsymbol{S}_t$ and $\boldsymbol{A}_t$ represent the state and action at time step $t$ defined in Section 2.3.2. Let the random vector $\boldsymbol{\Theta}_t = [\Theta_{1,t}, \Theta_{2,t}]^\top$ denote a learner's state $\boldsymbol{S}_t$ at time step $t$, which represents the latent traits in our study. Consider three sets of learning materials regarding the two-dimensional latent trait levels, that is, $\mathcal{A} = \{1, 2, 3\}$. Each set of learning materials contain contents with regards to different latent traits. Denote the change of the latent traits from time step $t$ to $t+1$ by $\Delta\boldsymbol{\Theta}_t = [\Delta\Theta_{1,t}, \Delta\Theta_{2,t}]^\top$. The probability of having $\Delta\boldsymbol{\theta} = [\Delta\theta_1, \Delta\theta_2]^\top$ transitioning from state $\boldsymbol{\theta} = [\theta_1, \theta_2]^\top$ to $\boldsymbol{\theta'} = [\theta'_1, \theta'_2]^\top$ can be represented as

$$\mathcal{P}(\boldsymbol{\theta'}|\boldsymbol{\theta}, l) = \mathbb{P}(\Delta\boldsymbol{\Theta}_t = \boldsymbol{\Delta\theta}|\boldsymbol{\Theta}_t = \boldsymbol{\theta}, \boldsymbol{A}_t = l), \tag{3.13}$$

where $l$ is the index of the set which the selected learning material belongs to. In the following notations, we only consider the set which the selected learning material belongs to, denoted as $l$. Assume $\theta_1, \theta_2 \in [0, 1]$, where the value of 0 indicates extremely low ability on the corresponding dimension and the value of 1 indicates the target ability.

In addition, under Assumption **A**3.1, we have $\Delta\theta_1 \in [0, 1-\theta_1]$ and $\Delta\theta_2 \in [0, 1-\theta_2]$. As we model the transition of the latent traits to be a continuous-state MDP, the change of $\Delta\theta_1$ and $\Delta\theta_2$ only depends on current latent trait $\boldsymbol{\theta}$ and the selected learning material $l$. Therefore, we let $\Delta\theta_1$ and $\Delta\theta_2$ follow Beta distributions such that $\Delta\theta_1 \sim Beta(1, g_1(\boldsymbol{\theta}, l))$, where $l \in \{1, 3\}$, and $\Delta\theta_2 \sim Beta(1, g_2(\Delta\theta_1, \boldsymbol{\theta}, l))$, where $l \in \{2, 3\}$. $\Delta\theta_2 = 0$ when $l = 1$ and $\Delta\theta_1 = 0$ when $l = 2$, which means the first set of materials only helps improving $\theta_1$ while the second set is only related to $\theta_2$. Parameters of $g_1(\boldsymbol{\theta}, l)$ and $g_2(\Delta\theta_1, \boldsymbol{\theta}, l)$ in the Beta distribution are calculated by

$$g_1(\boldsymbol{\theta}, l) = \begin{cases} 3 + 8\theta_1 - 0.2\theta_2, & l = 1 \\ 15 + 15\theta_1 - 0.4\theta_2, & l = 3 \end{cases} \tag{3.14}$$

and

$$g_2(\boldsymbol{\theta}, l) = \begin{cases} 10 - \theta_1 + 5\theta_2, & l = 2 \\ 20 - 28\theta_1 e^{-\frac{(\theta_1 - 0.6)^2}{0.3}} + 30\theta_2 - 0.3\Delta\theta_1, & l = 3. \end{cases} \tag{3.15}$$

An intuitive example is how a learner learns addition "+" and subtraction "–". A learning process usually takes a long time and thus a monotonic decreasing, zero-concentrated distribution is adopted to simulate the ability increase. In that case, each learning step will most likely lead to a small increase of the ability/abilities. Besides, in the distribution $Beta(1, b)$, the larger $b$ is, the more the curve approaches 0, which results in a higher chance in generating a smaller $\Delta\boldsymbol{\theta}$. It implies that a higher ability the learner has on either dimension, the harder for him/her to further improve the corresponding ability. Thus, $g_1(\boldsymbol{\theta}, l)$ and $g_2(\Delta\theta_1, \boldsymbol{\theta}, l)$ have positive coefficients in front of $\theta_1$ and $\theta_2$, respectively. Meanwhile, we assume that a higher ability on one dimension helps to increase the other dimension's ability, which results in a negative coefficient ahead of $\theta_2$ in $g_1(\boldsymbol{\theta}, l)$ and a negative coefficient ahead of $\theta_1$ in $g_2(\Delta\theta_1, \boldsymbol{\theta}, l)$. In particular, assume the third learning material contains contents related to both abilities, and especially helps learners with intermediate or high ability level of addition to improve further on subtraction. This assumption is included in calculating $g_2(\boldsymbol{\theta}, l)$ when $l = 3$ in equation (3.15). In addition, if the learner makes a big progress in mastering the ability of addition, there is a higher chance for the one to improve more on learning subtraction. Thus, the coefficient of $\Delta\theta_1$ in $g_2(\Delta\theta_1, \boldsymbol{\theta}, l)$ is negative which gives a curve that is less zero-concentrated as $\Delta\theta_1$ increases. Consequently, $\Delta\theta_2$ has a higher possibility in increasing more as $\Delta\theta_1$ is large. Note that the transition model is not required for adaptive learning system. The simulation gives an example in validating the data-driven, model-free deep Q-learning algorithm in discovering the optimal learning policy.

Estimation errors ranging from 1% to 15% are also added to estimated latent traits to evaluate their impacts on the adaptive learning system. Denote the estimation error vector by $\boldsymbol{e} = [e_1, e_2]^\top$, where $e_1$ and $e_2$ are generated by the same normal distribution such that $e_1, e_2 \sim \mathcal{N}(0, \sigma^2)$. As a result, 99.7% of $e_1, e_2$ lie in the range of $(-3\sigma, 3\sigma)$. In the simulation, the estimated latent traits are calculated by the sum of the true latent traits and the estimation errors, which are $[\theta_1 + e_1, \theta_2 + e_2]^\top$. For instance, if the standard deviation $\sigma$

is 0.03, the observation is $[\theta_1 + e_1, \theta_2 + e_2]^\top$, where $e_1, e_2 \sim \mathcal{N}(0, 0.03^2)$, and 99.7% of $e_1, e_2$ lie in the range of $(-0.09, 0.09)$.

Two simulations cases are studied. In the first case, the DQN is trained against actual learners whose abilities' changes follow the MDP with kernel distributions described above. In this case, it is presumed that the optimal learning policy can be trained on sufficient number of learners. The resulting optimal learning policy is compared with a heuristic learning policy, which selects the next learning material that can improve the not-fully-mastered ability, and a random learning policy which selects any material randomly from the set of three. The impact of different estimation errors is also assessed. In the second case, the DQN is trained against an estimated transition model learning that is obtained using a small group of learners. The resulting optimal learning policy is compared with that obtained by training against actual learners.

### 3.5.2 Simulation Study I

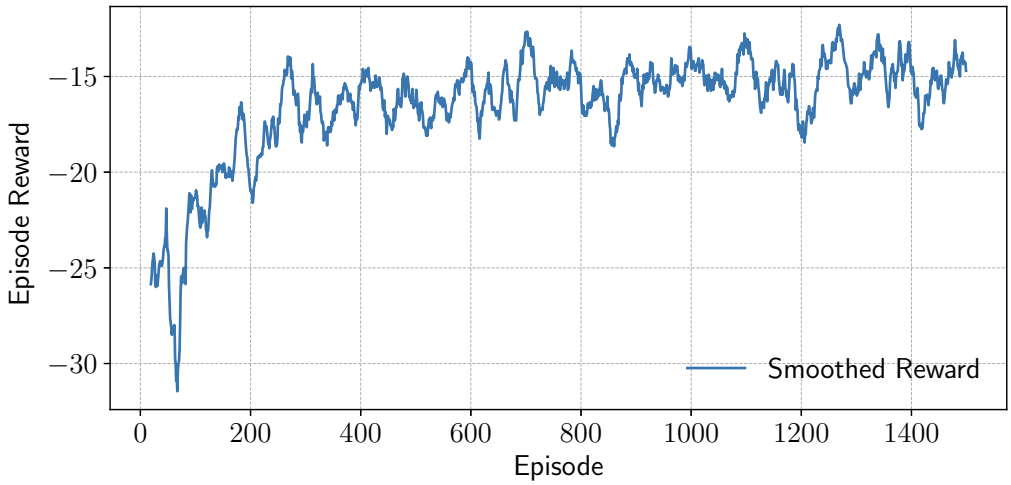


Figure 3.5: Smoothed rewards under the deep Q-learning algorithm.

Assume all learners are beginners on the two latent traits when using the adaptive learning system, i.e. $\boldsymbol{\Theta}_0 = [0, 0]^\top$. The DQN has two hidden layers, the first of which has 64 units and the second of which has 32 units. The DQN is trained against 2000 learners that are simulated according to the method discussed earlier, i.e. $E = 2000$. Other parameters are chosen as

Table 3.1: Mean and Standard Deviation (SD) of Rewards under DQN, Heuristic, and Random Learning Policies.

| Methods | DQN | Heuristic | Random |
|---|---|---|---|
| Reward mean | -13.49 | -21.55 | -24.85 |
| Reward SD | 4.59 | 4.76 | 5.59 |

follows: $\gamma = 0.9$, $\alpha = 6 \times 10^{-4}$, $\bar{\epsilon} = 1.0$, $\underline{\epsilon} = 0.1$, $\tau_\epsilon = 2000$, $M = 256$. The Adam optimization algorithm is adopted for the training of the DQN.

Figure 3.5 present the smoothed reward under the deep Q-learning algorithm across the first 1500 episodes with a smoothing window of 20. Since the DQN algorithm is adequately trained when the reward curve converges to certain value, it can be seen that the reward converges to $-15$ after 600 episodes, which indicates the optimal learning policy is found after the DQN is trained using 600 learners.

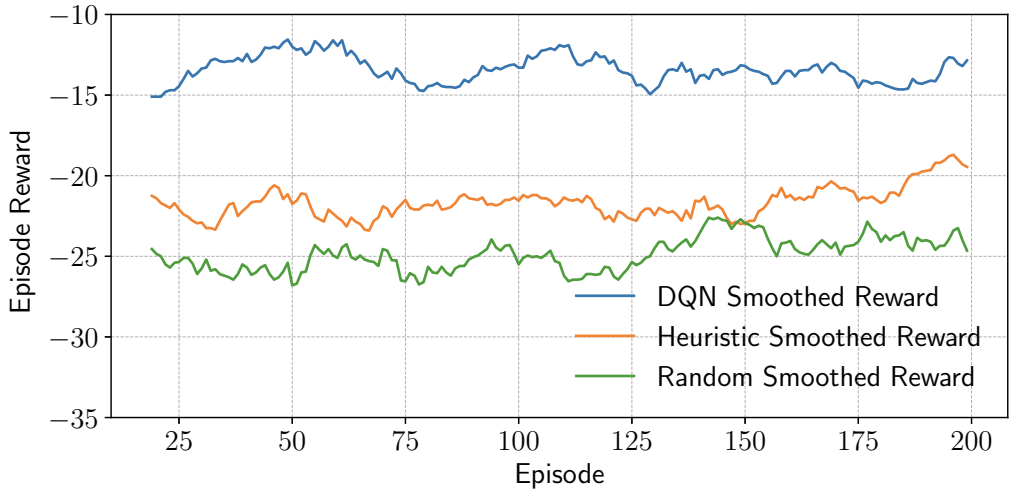


Figure 3.6: Smoothed rewards under DQN, heuristic, and random learning policies.

Figure 3.6 and Table 3.1 compare smoothed rewards across 200 new learners, labeled as episodes in Figure 3.6, with a smoothing window of 20 between the optimal learning policy found by the deep Q-learning algorithm after being trained in 2000 episodes—referred to as the DQN learning policy, the heuristic learning policy, and the random learning policy. The larger the reward is, the fewer learning steps a learner takes to fully master the two

latent traits, or in another word, the better the learning policy is. Clearly, the rewards obtained by the deep Q-learning algorithm have a higher mean and smaller standard deviation (SD) than those obtained by the heuristic learning policy and the random learning policy. These results show that the learning policy found by the deep Q-learning algorithm is much better than the other two.

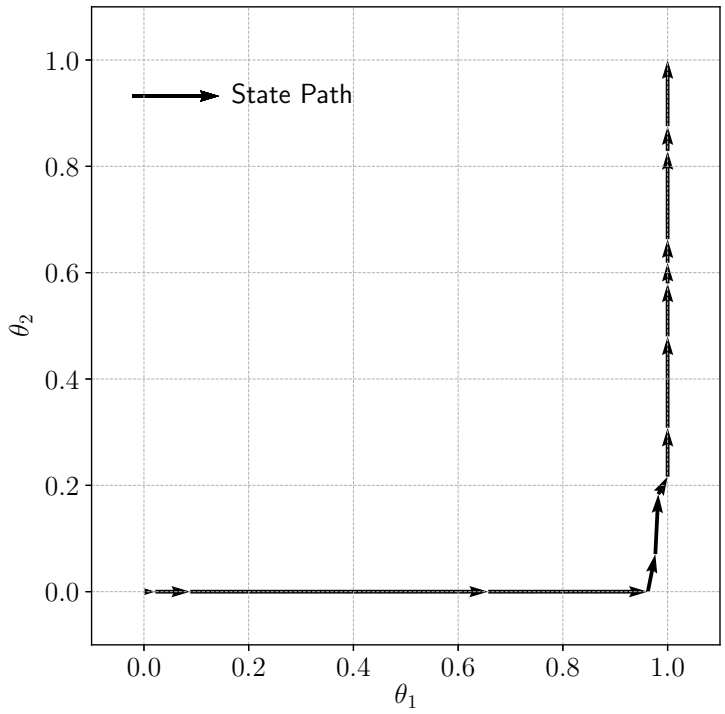


Figure 3.7: An example of state transition path with action sequence.[1]

Figure 3.7 presents an example of a state transition path that shows how the latent traits change with a sequence of actions taken under the DQN learning policy obtained without considering estimation error. Take the addition and subtraction test as an example. The first learning material is repeatedly selected to improve the learner's ability of addition at the beginning. Then the third material related to both addition and subtraction is selected. In the last few steps, the second learning material is chosen to further improve the learner's ability of subtraction.

[1]The action sequence in the example is 1, 1, 1, 1, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2.

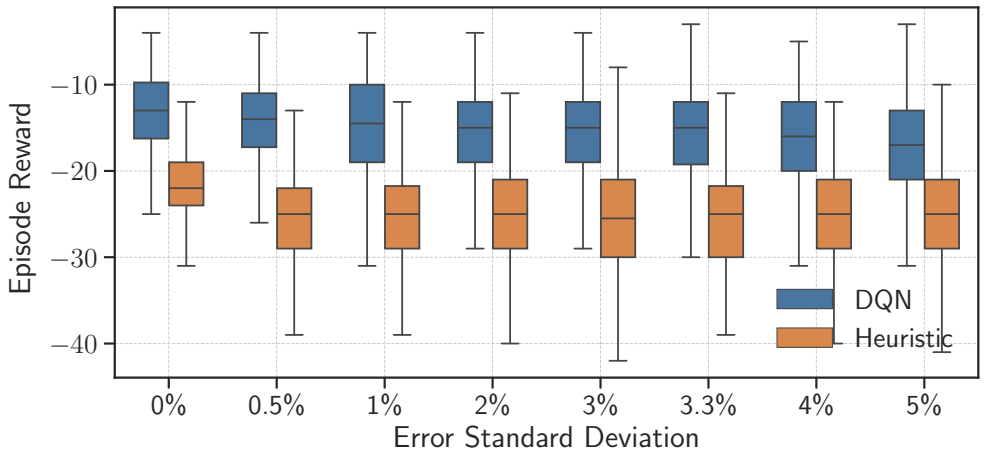


Figure 3.8: Comparison of rewards under DQN and heuristic learning policies with various estimation errors.

Figure 3.8 compares rewards under the DQN and the heuristic learning policies when estimation errors with various standard deviations ($\sigma$) exist. It shows that the mean rewards obtained by the DQN learning policy under various estimation errors are consistently higher than those of the heuristic learning policy when estimation errors exist. That is, the DQN learning policy still outperforms the heuristic learning policy even with the presence of estimation errors, which demonstrates that the deep Q-learning algorithm is reliable and stable in finding optimal learning policy with the presence of estimation errors.

### 3.5.3 Simulation Study II

Next, we show the performance of the adaptive learning system with a transition model estimator, which is represented using a neural network with one hidden layer that has 32 units. We tried different numbers of layers and different numbers of units in the simulation. Since a larger number of layers or more units in the hidden layer than required add more complexity in the model which will result in the over-fitting problem (potentially high train score but lower test score), while a smaller number of layers or less units may reduce the train score, we found that one hidden layer with 32 units can achieve both high train score and test score in this simulation study. The prediction accuracy indices are presented in Table 3.2. The train and test

Table 3.2: Accuracy of Transition Model Trained against Various Numbers of Learners.

| No. of learners | 10 | 20 | 30 | 40 | 50 | 100 | 150 | 200 | 2000 |
|---|---|---|---|---|---|---|---|---|---|
| Train Score | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| Test Score | 0.95 | 0.97 | 0.96 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| RMSE | 0.11 | 0.08 | 0.09 | 0.09 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |

scores are defined as the coefficient of determination in the training and test sets respectively, calculated by

$$1 - \frac{\sum_{i=1}^{H} ||\boldsymbol{s}^{(i)} - \hat{\boldsymbol{s}}^{(i)}||^2}{\sum_{i=1}^{H} ||\boldsymbol{s}^{(i)} - \overline{\boldsymbol{s}}||^2}, \tag{3.16}$$

where $\boldsymbol{s}$ is the true state, $\overline{\boldsymbol{s}}$ is average value of the true state, $\hat{\boldsymbol{s}}$ is the predicted state using previous state and the action taken, and $H$ is the number of the transitions. The best possible score is 1. The root mean square error (RMSE) is calculated by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{H} ||\boldsymbol{s}^{(i)} - \hat{\boldsymbol{s}}^{(i)}||^2}{H}}. \tag{3.17}$$

A DQN is trained on 2000 episodes against the estimated transition model that is fitted using a certain number of actual learners; the learning policy corresponding to this DQN is referred to as the virtual DQN learning policy. For the purpose of comparison, another DQN is trained on the same number of actual learners; the learning policy corresponding to this DQN is referred to as the actual DQN learning policy. Essentially, these two learning policies differ in the way how the same set of actual learners are utilized. The actual learners are simulated according to the method discussed in "Simulation Overview" section and are used to train the actual DQN learning policy directly. In contrast, these actual learners are used to first fit a transition model, which is then used to train the virtual DQN learning policy; this allows the virtual DQN learning policy to be trained over as many episodes as it needs. Figure 3.9 compares rewards obtained by the two DQN learning policies when various numbers of actual learners are utilized. It is shown that with no more than 200 actual learners, the utilization of the transition model can significantly improve the performance of the learning policy, generating much larger mean rewards compared than the algorithm without using the

transition model. When the number of learners is large enough, both two approaches found optimal learning policies and yield similar rewards.

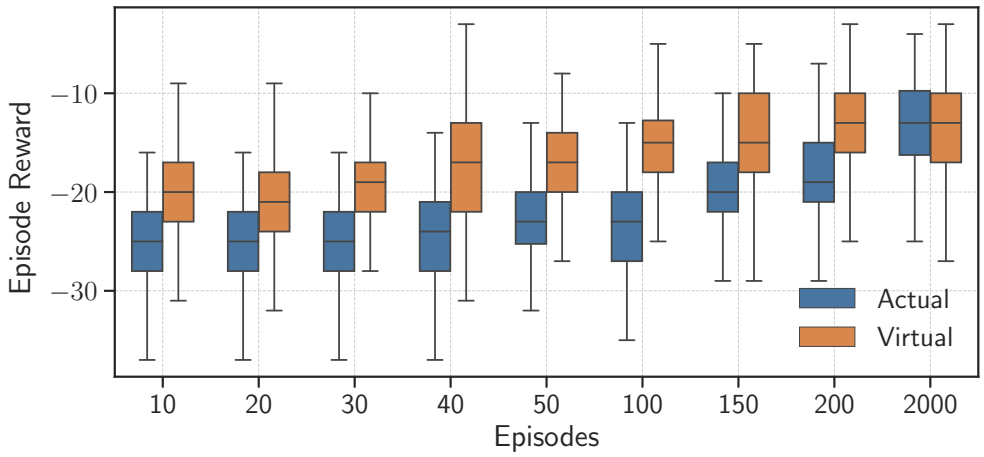


Figure 3.9: Comparison of rewards under actual and virtual DQN learning policies.

## 3.6 DISCUSSION

In this chapter, we developed an MDP formulation for an adaptive learning system by describing learners' latent traits as continuous instead of simply classifying learners as mastery or non-mastery of certain skills. The objective of the system is to improve learners' abilities to the prespecified target levels. We developed a deep Q-learning algorithm, which is a data-driven and model-free DRL algorithm that can effectively find the optimal learning policy from data on learners' learning process without knowing the transition model of the learner's latent traits. To cope with the challenge of insufficient state transition data, which may result in a poor performance of the deep Q-learning algorithm, we developed a transition model estimator that emulates the learner's learning process using neural networks, which can be used to further train the DQN and improve the its performance.

The two simulation studies presented in the chapter verified that the proposed methodology is very efficient in finding a good learning policy for adaptive learning systems without any help from a teacher. The optimal learning policy found by the DQN algorithm outperformed the heuristic and random methods with much higher rewards, or equivalently, much fewer learning

steps/cycles for learners to reach the target levels of all prespecified abilities. Particularly, with the aid of a transition model estimator, the adaptive learning system can find a good learning policy efficiently after training using a few learners.

# CHAPTER 4

# CONCLUDING REMARKS

In this dissertation, I have discussed several problems in adaptive learning systems and proposed feasible methods to solve these problems. In Chapter 2, I proposed two feasible and flexible methods to achieve the content balancing in a variable-length computerized classification test. These two methods handle content constraints much better than traditional content balancing methods including the maximum priority index method and the content weighted item selection index method, while our methods still generate similarly high classification accuracies compared to the traditional methods. Since the variable-length computerized adaptive test can estimate learners' abilities with as few items as possible, it is powerful and efficient in providing adaptive assessment experiences for learners. We can also directly adopt the proposed look-ahead content balancing method with a constant step size in the item selection procedure in variable-length computerized adaptive tests in addition to variable-length computerized classification tests.

In Chapter 3 and 4, models and methods were proposed to find the optimal learning policy for learners in various scenarios and under different conditions. In Chapter 3, I proposed the hierarchical learning model as a uniform framework to model both the attribute hierarchy structure and mastery levels of attributes in a flexible way. After the mastery levels of different attributes were estimated using CDMs, the transition process of learning path was modeled as an MDP. The data-driven Q-learning algorithm was used to find the optimal learning policy for each individual learner. We demonstrated that the Q-learning algorithm finds a better learning policy with fewer number of learning steps for learners to master all attributes compared to the heuristic method which is also a fairly good one in selecting individualized learning materials.

In Chapter 4, I further modeled learners' latent traits as continuous scales to provide more information for learners in a complicated learning and assessment environment. After the latent traits' transitions given learning materials were modeled as a continuous MDP, the data-driven deep Q-learning algorithm was applied to find the optimal learning policy. The optimal learn-

ing policy found by the deep Q-learning algorithm outperformed those found by the heuristic and random methods, with fewer number of learning steps taken by learners to reach the target levels of prespecified abilities. Furthermore, the transition model estimator was developed using neural networks to improve the learning policy found by the deep Q-learning algorithm when insufficient state transition data is available.

Several interesting research directions are worth exploring towards the content balancing methods in variable-length computerized classification tests and computerized adaptive tests. First, a variation of LA-CB method with an adaptive step size can be explored to deal with the problem of content balancing in variable-length computerized adaptive tests. Second, different stopping rules can be evaluated and optimally determined, and the LA-CB methods can be adjusted, especially the LA-CB-A method, based on the preferred stopping rule (Babcock and Weiss, 2009) to adapt to variable-length computerized classification tests and computerized adaptive tests.

In addition, several directions are possible for future research on finding the optimal learning policy in adaptive learning systems. First, to adopt and evaluate the data-driven algorithms, including the Q-learning algorithm and the DQN algorithm, and the transition model estimator through real data analysis with actual learners' data on an online learning platform. Second, because each group of learners assumes to follow a homogeneous MDP, further researches can be conducted to classify learners into groups before they use the adaptive learning system in order to find the optimal learning policy for each group. Particularly, dimensionality assessment methods can be explored to be used to classify learners at the first stage (Zhang and Stout, 1999; Zhang, 2013), in addition to using estimation methods to get learners' initial states. Third, different algorithms can be proposed to select the individualized learning materials that can maximize learners' immediate or future rewards (Manickam et al., 2017). Finally, the adaptive learning system here consists of a latent trait estimator which uses measurement models to estimate latent traits and a learning policy. Instead, some research construct the system assuming that learning materials influence learners' responses to test items directly, without the latent trait estimator incorporated (Lan et al., 2014; Lan and Baraniuk, 2016). As such, a learner learning process is modeled and traced directly and data-driven algorithms can be proposed to find the optimal learning policy.

## REFERENCES


Ackerman, T. A., Gierl, M. J., and Walker, C. M. (2003). Using multidimensional item response theory to evaluate educational and psychological tests. *Educational Measurement: Issues and Practice*, 22(3):37–51.

Babcock, B. and Weiss, D. J. (2009). Termination criteria in computerized adaptive tests: Variable-length cats are not biased. In *Proceedings of the 2009 GMAC conference on computerized adaptive testing*, volume 14.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*, volume 5. Athena Scientific Belmont, MA.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores*, pages 397–472.

Bolt, D., Chen, H., DiBello, L., Hartz, S., Henson, R., Roussos, L., Stout, W., and Templin, J. (2008). The arpeggio suite: software for cognitive skills diagnostic assessment [computer software and manual]. *St. Paul, MN: Assessment Systems*.

Brusilovsky, P. and Peylo, C. (2003). Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education (IJAIED)*, 13:159–172.

Caicedo, J. C. and Lazebnik, S. (2015). Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2496.

Chang, H.-H. (2015). Psychometrics behind computerized adaptive testing. *Psychometrika*, 80(1):1–20.

Chang, H.-H. and Ying, Z. (1999). A-stratified multistage computerized adaptive testing. *Applied Psychological Measurement*, 23(3):211–222.

Chen, J. and de la Torre, J. (2013). A general cognitive diagnosis model for expert-defined polytomous attributes. *Applied Psychological Measurement*, 37(6):419–437.

Chen, S.-Y. and Ankenman, R. D. (2004). Effects of practical constraints on item selection rules at the early stages of computerized adaptive testing. *Journal of Educational Measurement*, 41(2):149–174.

Chen, Y., Culpepper, S. A., Chen, Y., and Douglas, J. (2018a). Bayesian estimation of the dina q matrix. *Psychometrika*, 83(1):89–108.

Chen, Y., Culpepper, S. A., Wang, S., and Douglas, J. (2018b). A hidden markov model for learning trajectories in cognitive diagnosis with application to spatial rotation skills. *Applied psychological measurement*, 42(1):5–23.

Chen, Y., Li, X., Liu, J., and Ying, Z. (2018c). Recommendation system for adaptive learning. *Applied psychological measurement*, 42(1):24–41.

Cheng, Y. and Chang, H.-H. (2009). The maximum priority index method for severely constrained item selection in computerized adaptive testing. *British Journal of Mathematical and Statistical Psychology*, 62(2):369–383.

Cheng, Y., Chang, H.-H., and Yi, Q. (2007). Two-phase item selection procedure for flexible content balancing in cat. *Applied Psychological Measurement*, 31(6):467–482.

Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.

de la Torre, J. (2009). Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 34(1):115–130.

de la Torre, J. (2011). The generalized dina model framework. *Psychometrika*, 76(2):179–199.

DiBello, L., Roussos, L., and Stout, W. (2007). Review of cognitively diagnostic assessment and a summary of psychometric models. cr rao, & s. sinharay (eds.), handbook of statistics, vol. 26: Psychometrics (pp. 970–1030).

Eggen, T. and Straetmans, G. (2000). Computerized adaptive testing for classifying examinees into three categories. *Educational and Psychological measurement*, 60(5):713–734.

Eggen, T. J. (1999). Item selection in adaptive testing with the sequential probability ratio test. *Applied Psychological Measurement*, 23(3):249–261.

Embretson, S. (1984). A general latent trait model for response processes. *Psychometrika*, 49(2):175–186.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., et al. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 11(3-4):219–354.

George, A. C., Robitzsch, A., Kiefer, T., Groß, J., and Ünlü, A. (2016). The r package cdm for cognitive diagnosis models. *Journal of Statistical Software*, 74(2):1–24.

Geramifard, A., Walsh, T. J., Tellex, S., Chowdhary, G., Roy, N., How, J. P., et al. (2013). A tutorial on linear function approximators for dynamic programming and reinforcement learning. *Foundations and Trends® in Machine Learning*, 6(4):375–451.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Green, B. F., Bock, R. D., Humphreys, L. G., Linn, R. L., and Reckase, M. D. (1984). Technical guidelines for assessing computerized adaptive tests. *Journal of Educational Measurement*, 21(4):347–360.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3389–3396. IEEE.

Hambleton, R. K. and Swaminathan, H. (2013). *Item response theory: Principles and applications*. Springer Science & Business Media.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality*. PhD thesis, ProQuest Information & Learning.

Henson, R. A., Templin, J. L., and Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191.

Huo, Y. (2009). *Variable-length computerized adaptive testing: adaptation of the a-stratified strategy in item selection with content balancing*. University of Illinois at Urbana-Champaign.

Junker, B. W. and Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Karelitz, T. M. (2004). *Ordered category attribute coding framework for cognitive assessments*. PhD thesis, University of Illinois at Urbana-Champaign Champaign, IL.

Kaya, Y. and Leite, W. L. (2017). Assessing change in latent skills across time with longitudinal cognitive diagnosis modeling: An evaluation of model performance. *Educational and psychological measurement*, 77(3):369–388.

Kingsbury, G. G. and Weiss, D. J. (1983). A comparison of irt-based adaptive mastery testing and a sequential mastery testing procedure. In *New horizons in testing*, pages 257–283. Elsevier.

Lan, A. S. and Baraniuk, R. G. (2016). A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429.

Lan, A. S., Studer, C., and Baraniuk, R. G. (2014). Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 452–461. ACM.

Leighton, J. P., Gierl, M. J., and Hunka, S. M. (2004). The attribute hierarchy method for cognitive assessment: A variation on tatsuoka's rule-space approach. *Journal of educational measurement*, 41(3):205–237.

Leung, C.-K., Chang, H.-H., and Hau, K.-T. (2000). *Content balancing in stratified computerized adaptive testing designs*. ERIC Clearinghouse.

Li, F., Cohen, A., Bottge, B., and Templin, J. (2016). A latent transition analysis model for assessing change in cognitive skills. *Educational and Psychological Measurement*, 76(2):181–204.

Li, X., Xu, H., Zhang, J., and Chang, H.-h. (2018). Optimal hierarchical learning path design with reinforcement learning. *arXiv preprint arXiv:1810.05347*.

Lin, C.-J. (2011). Item selection criteria with practical constraints for computerized classification testing. *Educational and Psychological Measurement*, 71(1):20–36.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pages 157–163. Elsevier.

Liu, J., Xu, G., and Ying, Z. (2012). Data-driven learning of q-matrix. *Applied psychological measurement*, 36(7):548–564.

Lord, F. (1980). Application of item response theory to practical testing problems. hillsdale, nj, lawrence erlbaum ass.

Lord, F. M., Novick, M. R., and Birnbaum, A. (1968). Statistical theories of mental test scores. 1968. *Reading: Addison-Wesley Google Scholar*.

Makransky, G. and Glas, C. A. (2014). An automatic online calibration design in adaptive testing. *Journal of Applied Testing Technology*, 11(1):1–20.

Manickam, I., Lan, A. S., and Baraniuk, R. G. (2017). Contextual multi-armed bandit algorithms for personalized learning action selection. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 6344–6348. IEEE.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, 64(2):187–212.

Masters, G. N. (1982). A rasch model for partial credit scoring. *Psychometrika*, 47(2):149–174.

McGlohen, M. and Chang, H.-H. (2008). Combining computer adaptive testing technology with cognitively diagnostic assessment. *Behavior Research Methods*, 40(3):808–821.

Means, B., Toyama, Y., Murphy, R., Bakia, M., and Jones, K. (2009). Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies.

Melo, F. S. and Ribeiro, M. I. (2007). Q-learning with linear function approximation. In *International Conference on Computational Learning Theory*, pages 308–322. Springer.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Mulaik, S. (1972). A mathematical investigation of some multidimensional rasch models for psychological tests. In *Annual Meeting of the Psychometric Society, Princeton, NJ*.

Muraki, E. (1992). A generalized partial credit model: Application of an em algorithm. *ETS Research Report Series*, 1992(1):i–30.

Muthén, L. and Muthén, B. (1998). Mplus. *The comprehensive modelling program for applied researchers: user's guide*, 5.

Norris, J. R. (1998). *Markov chains*. Number 2. Cambridge university press.

Parshall, C. G. (2002). *Practical considerations in computer-based testing*. Springer Science & Business Media.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. *Copenhagen: Danish Institute for Educational Research*.

Reckase, M. D. (1972). Development and application of a multivariate logistic latent trait model.

Reddy, S., Levine, S., and Dragan, A. (2017). Accelerating human learning with deep reinforcement learning. In *NIPS'17 Workshop: Teaching Machines, Robots, and Humans*.

Roussos, L. A., Templin, J. L., and Henson, R. A. (2007). Skills diagnosis using irt-based latent class models. *Journal of Educational Measurement*, 44(4):293–311.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika monograph supplement*.

Spray, J. A. and Reckase, M. D. (1996). Comparison of sprt and sequential bayes procedures for classifying examinees into two categories using a computerized test. *Journal of Educational and Behavioral Statistics*, 21(4):405–414.

Sternberg, R. J. and Ben-Zeev, T. (1996). *The nature of mathematical thinking*. Routledge.

Studer, C. (2012). Incorporating learning over time into the cognitive assessment framework. *Unpublished PhD, Carnegie Mellon University, Pittsburgh, PA*.

Su, Y.-H. (2015). The performance of the modified multidimensional priority index for item selection in variable-length mcat. In *Quantitative psychology research*, pages 89–97. Springer.

Su, Y.-H. (2016). A comparison of constrained item selection methods in multidimensional computerized adaptive testing. *Applied psychological measurement*, 40(5):346–360.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Swygert, K. (2002). Practical considerations in computer-based testing.

Sympson, J. and Hetter, R. (1985). Controlling item-exposure rates in computerized adaptive testing. In *Proceedings of the 27th annual meeting of the Military Testing Association*, pages 973–977.

Sympson, J. B. (1978). A model for testing with multidimensional items. In *Proceedings of the 1977 computerized adaptive testing conference*, number 00014. University of Minnesota, Department of Psychology, Psychometric Methods.

Tang, X., Chen, Y., Li, X., Liu, J., and Ying, Z. (2019). A reinforcement learning approach to personalized learning recommendation systems. *British Journal of Mathematical and Statistical Psychology*, 72(1):108–135.

Templin, J., Henson, R. A., et al. (2010). *Diagnostic measurement: Theory, methods, and applications*. Guilford Press.

Templin, J. L. (2004). Generalized linear mixed proficiency models. *Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign*.

Templin, J. L. and Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological methods*, 11(3):287.

Thissen, D. and Mislevy, R. J. (2000). Testing algorithms. *Computerized adaptive testing: A primer*, 2:101–133.

Thompson, N. A. (2009). Item selection in computerized classification testing. *Educational and Psychological Measurement*, 69(5):778–793.

Thompson, N. A. and Prometric, T. (2007). A practitioner's guide for variable-length computerized classification testing. *Practical Assessment Research & Evaluation*, 12(1):1–13.

Thrun, S. and Schwartz, A. (1993). Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum*.

Tonidandel, S., Quiñones, M. A., and Adams, A. A. (2002). Computer-adaptive testing: The impact of test characteristics on perceived performance and test takers' reactions. *Journal of Applied Psychology*, 87(2):320.

Tseng, F.-L. and Hsu, T.-C. (2001). Multidimensional adaptive testing using the weighted likelihood estimation: a comparison of estimation methods. In *Annual meeting of NCME, Seattle*.

Tu, D., Wang, S., Cai, Y., Douglas, J., and Chang, H.-H. (2018). Cognitive diagnostic models with attribute hierarchies: Model estimation with a restricted q-matrix design. *Applied Psychological Measurement*, page 0146621618765721.

von Davier, M. (2006). Multidimensional latent trait modelling (mdltm)[software program]. *Princeton, NJ: Educational Testing Service*.

Wainer, H., Dorans, N. J., Flaugher, R., Green, B. F., and Mislevy, R. J. (2000). *Computerized adaptive testing: A primer*. Routledge.

Wald, A. (1973). *Sequential analysis*. Courier Corporation.

Wang, C. (2015). On latent trait estimation in multidimensional compensatory item response models. *Psychometrika*, 80(2):428–449.

Wang, S., Yang, Y., Culpepper, S. A., and Douglas, J. A. (2018). Tracking skill acquisition with cognitive diagnosis models: A higher-order, hidden markov model with covariates. *Journal of Educational and Behavioral Statistics*, 43(1):57–87.

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3):427–450.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, Cambridge.

Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement*, 6(4):473–492.

Weiss, D. J. and Kingsbury, G. (1984). Application of computerized adaptive testing to educational problems. *Journal of Educational Measurement*, 21(4):361–375.

Whitely, S. E. (1980). Multicomponent latent trait models for ability tests. *Psychometrika*, 45(4):479–494.

Xu, H., Sun, H., Nikovski, D., Kitamura, S., Mori, K., and Hashimoto, H. (2019). Deep reinforcement learning for joint bidding and pricing of load serving entity. *IEEE Transactions on Smart Grid*, pages 1–1.

Xu, J., Xing, T., and Van Der Schaar, M. (2016). Personalized course sequence recommendations. *IEEE Transactions on Signal Processing*, 64(20):5340–5352.

Yao, L. (2013). Comparing the performance of five multidimensional cat selection procedures with different stopping rules. *Applied Psychological Measurement*, 37(1):3–23.

Zhang, J. (2013). A procedure for dimensionality analyses of response data from various test designs. *Psychometrika*, 78(1):37–58.

Zhang, J. and Stout, W. (1999). The theoretical detect index of dimensionality and its application to approximate simple structure. *Psychometrika*, 64(2):213–249.

Zhang, J., Xie, M., Song, X., and Lu, T. (2011). Investigating the impact of uncertainty about item parameters on ability estimation. *Psychometrika*, 76(1):97–118.

Zhang, S. and Chang, H.-H. (2016). From smart testing to smart learning: how testing technology can assist the new generation of education. *International Journal of Smart Technology and Learning*, 1(1):67–92.